Les objets en PHP 5

Jean-Ferdinand Susini

21 mars 2011

basé sur les transparents deJ.-F. Dazy et J.-F. Berger

CONSERVATOIRE NATIONAL
DES ARTS ET METIERS

Cnam
CONSERVATOIRE NATIONAL
DES ARTS ET METIERS

Modèle de programmation

- Un programme PHP est habituellement structuré selon le modèle procédural : le programmeur définit des données structurées et an parallèle les traitements de ces données sous forme de procédures (les fonctions PHP)

```
function nomFonction($arg1, $arg2 ...){
  Instructions
}
```

- Les fonctions PHP sont récursives et admettent un nombre quelconque d'arguments. À partir d'un certain rang, on peut fixer des valeurs par défaut aux arguments. La valeur de retour est unique et peut avoir un type quelconque.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Exemples

```
function testArgs($i,$j="truc"){
 echo "testing testArg($i, $j): ";
 echo "num args = ".func_num_args()."<br>";
 print_r($i);
 echo "  ";
 print_r($j);
 echo " - > args2 = ".func_get_arg(2);
 echo "<br>";
}

testArgs();
testArgs(1);
testArgs(1,"o");
testArgs(1,"o","t");
```

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Modèle de programmation

- Le passage par référence des paramètres est supporté par l'utilisation de valeurs de type référence '&'

- Les fonctions peuvent retourner des références

```
function &getReference(){
  return $ref;
}
```

- On peut définir des références de fonctions à l'aide de leur nom "les fonctions variables"

```
function foo(){ echo “foo() <br/> \n";}
$func='foo';
$func(); //Appel de foo() !!!!
```

Cnam
CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Les objets : encapsulation

- On regroupe dans la même entité des **données structurées** et les **fonctions** qui les manipulent
- Les données sont généralement appelées **attributs** et les procédures de traitement sont appelées **méthodes**
- Idéalement, pour accéder aux données on passe par les méthodes => notion d'interface : isoler l'implémentation d'un objet à la manière d'un type de donnée abstrait
- Deux modèles coexistent PHP4 (deprecated) et PHP5. On présente le modèle de PHP5 (meilleures performances et modèle plus riche)

Cnam
CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Exemples

```
class Voiture
{
  private $nb_tour = 5;
  private $rayonRoue = 0.2;

  public function getSpeed(){
    return $this->nb_tour * $this->rayonRoue * 2 * M_PI;
  }
}

$v = new Voiture();
echo "vitesse voitue = ".$v->getSpeed();
```

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Manipulation d’objets

- Les **objets** sont manipulés par l’intermédiaire de **références**
- Un attribut d’un objet peut-être accédé par la notation suivante :

```
référence_objet->nom_du_champ
```

- Une méthode d’un objet peut être appelée par la notation suivante :

```
référence_objet->nom_de_la_méthode(valeurs)
```

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Modèle à héritage de classe

- Le **type d’un objet** est défini par une **classe** (description d’un objet : attributs et méthodes)
- déclaration :

```
[abstract|final] class Nom [extends ClasseMère]
                           [implements Interface[, Interface]]
{
//déclaration d’un attribut
    [public|protected|private] [static] $att [= valeur];
//déclaration d’une constante
    const NOM = valeur ;
//déclaration d’une méthode
    [public|protected|private] [abstract|final] [static]
                            function f(args..)[ { ... } | ;]
}
```

- La classe est une “usine à objets” et peut posséder des attributs partagés entre toutes les instances

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Mots clés et variables réservées (méthodes)

- La variable `$this` référence l'instance courante sur laquelle on exécute une méthode d'instance (lecture seule).
- On accède aux attributs et aux méthodes par la construction syntaxique suivante :
  `$this->nomAttributOuMethode // pas de $ devant un attribut`
- Les variables et méthodes de classe sont appelées en utilisant l'opérateur ::
  `nomClasse::$nomAttribut nomClasse::nomMethode()`
- `$this` n'est pas défini pour les méthodes de classe. Pour les attributs statiques `self::$attrib`
- Les constantes dans une classe sont accessibles par :
  `self::nomConstante`

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

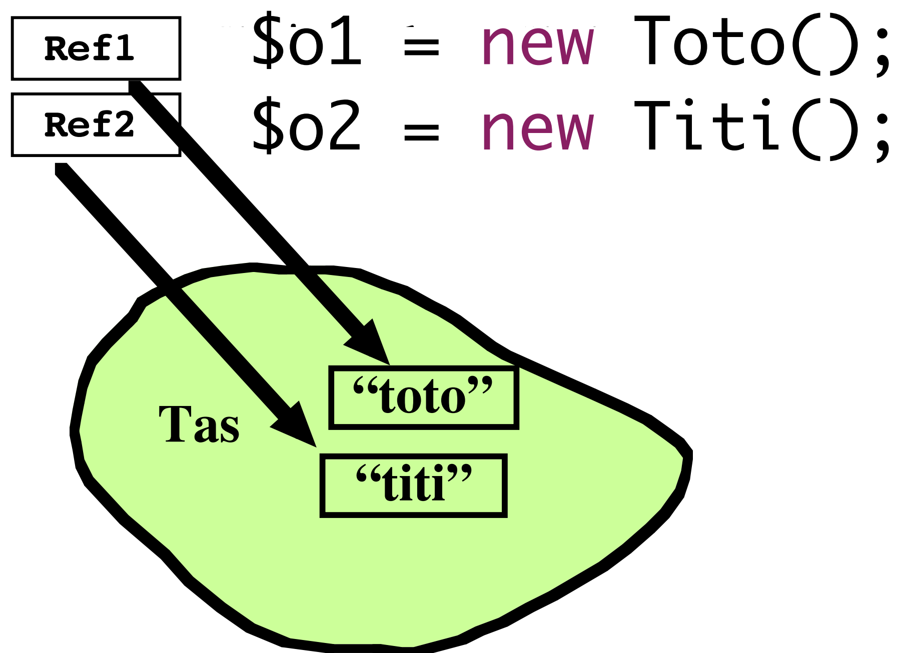
Instanciation d'objets

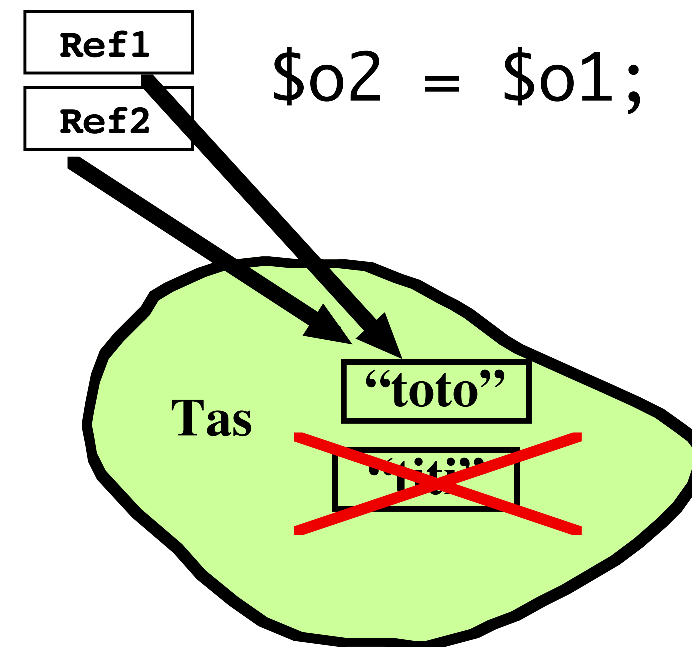
- Création d'objets à partir d'une classe = **instanciation** on utilise le mot clé `new` et on invoque implicitement un constructeur (fonction magique unique)
  `[public|protected|private] function __construct() {...}`
- Une référence est crée sur la représentation de l'objet en mémoire centrale, l'identifiant de manière unique. La référence `NULL` est retourné si l'objet ne peut être créé

```
class Test{
  private $nom;
  public function __construct($nom){
    $this->nom = $nom;echo "Say Hello to $this->nom";
  }
}
$refToto = new Test("moi");
```

Cnam
CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Référence à un objet





CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Gestion de la mémoire et objets

- La référence sur un objet en mémoire centrale doit être affectée à une variable. Lorsqu'il n'y a plus de référence sur un objet dans un programme, celui-ci est libéré.

- La libération d'un objet entraîne l'exécution d'une fonction magique : le destructeur

```
[public|protected|private] function __destruct(){...}
```

- La référence sur un objet en mémoire centrale doit être conservée dans une variable :

```
class Test{
  public $nom;
  public function __destruct(){
    echo "Say Good bye to $this->nom";
  }
}
new Test(); //exécution immédiate du destructeur
```

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Les modificateurs d’accès

- `private` : l'attribut (resp. la méthode) n’est visible qu’à l’intérieur de la classe (pas d’étanchéité entre instances).
- `protected` : l'attribut (resp. la méthode) n’est visible qu’à l’intérieur de la classe et dans les classes dérivées.
- `public` : l'attribut (resp. la méthode) appartient à l’interface utilisateur de la classe, ce sont les seuls éléments visibles pour un utilisateur (extérieur) de la classe. C'est l'accès par défaut => le mot clé `public` est optionnel
- En cas de redéfinition, on ne peut pas restreindre l’accessibilité (exemple `public` -> `private`)

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Réutilisation de code : héritage

- Caractéristique d’un langage OO ; mécanisme de réutilisation et de spécialisation
- Héritage simple en PHP (arborescences de classes)
- redéfinition de méthodes dans les classes filles et à travers la fonction magique `__call` .
- pas de redéfinition des variables
- mise en œuvre avec le mot clé `extends`
- référence à la classe parente avec le mot clé `parent` (réutilisation)

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Les interfaces

- Réponse partielle au problème de l'héritage multiple, déclaration :

```
interface NomDInterface
{...}
```

- **Constituée de signatures de fonctions publiques** et de **constantes de classes** (`public static`)
- Une classe peut implémenter une ou plusieurs interfaces mot clé `implements`
- Représente un contrat entre un objet et celui qui le manipule (implémente toutes les méthodes)

Cnam
CONSERVATOIRE NATIONAL DES ARTS ET METIERS

PHP5, Comparaison d'objets: ==, ===

- l'opérateur de comparaison == compare les objets de manière "classique" : 2 objets sont égaux s'ils ont les mêmes attributs avec les mêmes valeurs, et s'ils sont des instances de la même classe.
- L'utilisation de l'opérateur d'identité === compare les références :il retourne TRUE si les références sont égales i.e. si elles font référence à un même objet en mémoire centrale.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Clonage

- Pour copier un objet on utilise le mot-clé `clone`

```
$copy = clone $obj;
```

ou

```
$copy = clone($obj);
```

- ATTENTION : clone effectue une copie superficielle de toutes les propriétés de l'objet. Tous les attributs qui sont des références à d'autres variables demeureront des références. Si une méthode magique __clone() est définie dans la classe alors la méthode__clone() du nouvel objet sera appelée pour permettre à chaque propriété qui doit l'être d'être modifiée.

CNAM CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Sérialisation

- C'est la possibilité de transformer un objet PHP dans un format structuré ou non
- Solution simple : transformation en chaîne de caractères à l'aide de la fonction magique `__tostring` (appelée au moment de la conversion en chaîne de caractères)
- Solution préférable on utilise les fonctions `serialize()` et `unserialize()`. Qui invoquent les fonctions magiques `__sleep` et `__wakeup` qui disposent d'implantation par défaut.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Sérialisation

- `__sleep` : permet de préciser les attributs à sérialiser. La fonction `serialize()` vérifie si la classe implémente `__sleep`. Si c'est le cas, elle l'exécute avant toute linéarisation. Elle peut nettoyer l'objet et elle retourne un tableau avec les noms de toutes les variables de l'objet qui doivent être linéarisées.

- Le but de `__sleep` est aussi de fermer toutes les connexions aux bases de données que l'objet peut avoir ouverte, de valider les données en attente ou d'effectuer des tâches de nettoyage. Cette fonction est utile pour les très gros objets qui n'ont pas besoin d'être sauvegardés en totalité.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Déserialisation

- `__wakeup()`, si définie, est appelée après dé-sérialisation pour recréer les attributs qui n'ont pas été sérialisés. La fonction `unserialize()` vérifie la présence d'une fonction magique `__wakeup` dans la définition de la classe. Si elle est présente, cette fonction permet de reconstruire toutes les ressources que l'objet possède. Le but de `__wakeup` est aussi de rétablir toute connexion à une base de données qui aurait été perdue durant la linéarisation et d'effectuer les tâches de réinitialisation.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Le chargement automatique

- PHP permet le chargement automatique de classes à la demande (équivalent des class loader en Java par exemple)
- L'interprète PHP gère une pile de fonctions, dites *de chargement automatique* qui sont appelées successivement lorsqu'une une classe est manipulée par un programme alors que celle-ci n'a pas encore été définie
- une fonction de chargement automatique est une fonction PHP écrite par le développeur d'application lui permettant de charger à la demande un fichier contenant la définition d'une classe non encore définie. Cette fonction reçoit comme argument le nom de la classe. En éventuel second paramètre fixe les extensions de fichiers à considérer

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Le chargement automatique

- La pile est gérée par : spl_autoload_register, spl_autoload_functions, spl_autoload_unregister
- Exemple de mise en œuvre :

```
class MyClassLoader
{
  public static function loadClass($class){
    echo "class name to load $class";
    require "CLASS_$class.php";
  }
}
spl_autoload_register('MyClassLoader::loadClass');

$toto = new Toto;
```

Exceptions

- Gestion privilégiée des erreurs. Identification bien claire des erreurs et il y a un “déroutage” du fil d’exécution en cas d’erreur et suivit d’un chemin spécifique

- L’exception est un objet (ErrorException : Gestion des erreurs qui peuvent elles aussi être interceptées)

```
throw new Exception ('une erreur');
```

- Attraper une exception

```
try{
...
}
catch (Exception $e){
    echo 'un message : '. $e->getMessage();
}
```

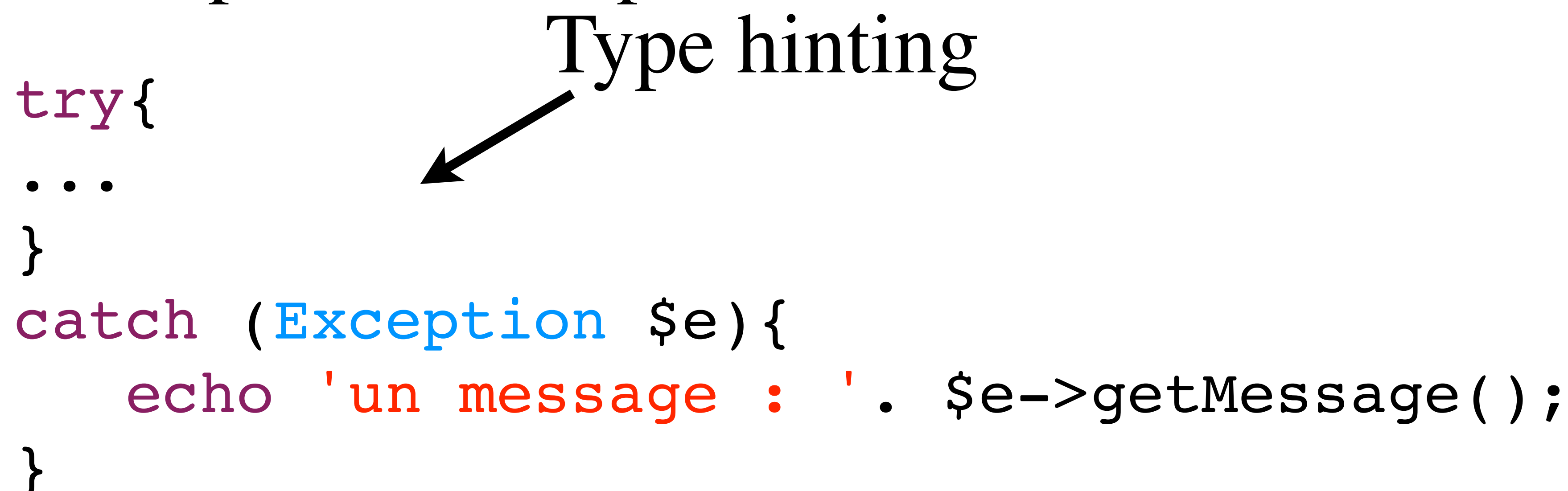
Exceptions

- Gestion privilégiée des erreurs. Identification bien claire des erreurs et il y a un “déroutage” du fil d’exécution en cas d’erreur et suivit d’un chemin spécifique

- L’exception est un objet (ErrorException : Gestion des erreurs qui peuvent elles aussi être interceptées)

```
throw new Exception ('une erreur');
```

- Attraper une exception



CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Introspection

- L'opérateur **instanceof** permet dans un code PHP de découvrir si un objet est une instance d'un classe précise

```
if( $o instanceof A){...}
if( $o instanceof $cl){...}
if( $o1 instanceof $o2){...}
```

- l'opérateur répond vrai si l'objet est de la classe spécifiée ou d'une sous classe
- l'operateur permet aussi de tester l'implantation d'interfaces. Mais dans ce dernier cas seule la dernière forme ne fonctionne pas

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Introspection

- Les classes ReflectionClass et ReflectionObject permettent d'inspecter à l'exécution le contenu d'une classe :

```
$inspect = new ReflectionClass('Toto');
$inspect->hasProperty('x');
$inspect->hasMethod('m');
$inspect->hasConstant('c');
```

- La récupération d'information est possible par les méthodes getMethod, getProperty, getConstant, getMethods, getProperties, getConstants,…
- Des méthodes d'inspection sur l'héritage : getParentClass, isSubclassOf, isInstanciable, isAbstract, isFinal, is Interface, implementsInterface, getInterfaces, …

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Introspection

- D'autres outils : getFileName, isInternal, isUserDefined, getStartLine, getEndLines, getDocComment,…
- ReflectionProperty : cette classe permet de manipuler un attribut d'une classe. getName, getValue, isPublic, isProtected, …
- RelflectionMethod : cette classe permet de manipuler une méthode d'une classe. isPublic, isPrivate, isStatic, isFinal, isAbstract, isConstructor, invoke,…