L3 MIASHS – Année 2018-2019
Programmation web
TD n° 1 : Bases de PHP objet
J. Darmont (http://eric.univ-lyon2.fr/jdarmont/), 05/02/19

On souhaite programmer en PHP une page web dont l'aspect sera le suivant.

http://eric.univ-lyon2.fr/jdarmont/docs/web/td1.php

Préliminaires

1. Lancer l'application XAMPP (Control Panel), qui comprend un serveur web Apache et un serveur MariaDB. Démarrer le serveur Apache si nécessaire.
2. Créer le squelette d'une page web classique (en-tête, corps, etc.). La sauvegarder sous le nom td1.php dans un sous-répertoire à votre nom (à créer) du répertoire C:\xampp\htdocs.

Hello world!

1. Créer dans td1.php une classe HelloWord contenant un attribut privé $n (lui affecter une valeur par défaut quelconque) et une méthode publique afficherN() qui affiche dans un paragraphe HTML la chaîne de caractère « Valeur de $n = » suivie de la valeur de l'attribut $n.
2. Créer un objet $h de classe HelloWorld et appeler la méthode afficherN() pour $h.
3. Le résultat peut être visualisé depuis un navigateur web à l'URL suivante. http://localhost/votreRépertoire/td1.php.
4. Est-il possible d'afficher la valeur de l'attribut $n sans passer par la méthode afficherN() ?

Tableaux et boucles

1. Créer une classe Tableau contenant un attibut privé $t, un constructeur qui affecte à $t un paramètre tableau $param_tab, et une méthode publique afficheV() qui affiche toutes les valeurs de $t dans une liste numérotée à l'aide d'une boucle « pour tout élément ». Créer un objet $tscal de classe Tableau avec en paramètre un tableau scalaire de chaînes de caractères avec quelques valeurs (ex. chaîne0, chaîne1, chaîne2…). Tester la création et l'affichage avec la méthode afficheV() du tableau $tscal.
2. Ajouter à la classe Tableau une méthode publique afficheKV() qui affiche dans une liste à puces les couples clés-valeurs de $t à l'aide d'une boucle « pour tout élément ». Créer un objet $tassoc de classe Tableau avec en paramètre un tableau associatif contenant le menu du restaurant universitaire, chaque « indice » représentant un jour de la semaine et chaque valeur stockée dans le tableau le nom du plat de résistance. Tester la création et l'affichage avec la méthode afficheKV() du tableau $tAssoc.
3. Créer une classe Matrice qui hérite de la classe Tableau, son constructeur (qui hérite du constructeur de Tableau) et une méthode publique sommeMat() qui retourne la somme des valeurs contenues dans la matrice $t à l'aide de deux boucles « pour tout élément » imbriquées. Créer un objet $mat de classe Matrice avec en paramètre un tableau numérique à deux dimensions (voir exemple ci-dessous). Tester la création de la matrice $mat et l'affichage du résultat de la méthode sommeMat() appliquée à $mat.

$$\begin{pmatrix} 1 & 3 & 5 \\ 9 & 0 & 2 \end{pmatrix}$$

4. Que faut-il changer dans la classe Tableau pour que cela fonctionne ?

Boucles et tests

1. Créer une classe Triangle contenant un attribut privé $taille, un constructeur qui affecte un paramètre entier $param_taille à $taille, et une méthode publique affiche() qui affiche (surprise !) un triangle rectangle constitué de caractères *. Notons que $taille est le nombre de lignes dans l'exemple ci-dessous. Créer un objet $tri1 de classe Triangle et de taille 5 et appeler la méthode affiche() pour $tri1.

   Ex. ($tri1)
   ```
   *
   **
   ***
   ****
   *****
   ```

2. Ajouter dans la methode affiche(), avant l'affichage du triangle, des tests sur la valeur de l'attribut $taille. Si elle est supérieure à 20, afficher un message « Triangle trop grand » et ne pas afficher le triangle. Si elle est inférieure ou égale à 0, lui attribuer une valeur de 10. Créer trois objets $tri2, $tri3 et $tri4 de classe Triangle et de taille 5, 50 et -5, respectivement. Les afficher.
3. Dans la methode affiche(), calculer et retourner le nombre d'étoiles (*) dont est constitué le triangle affiché. Créer un objet $tri5 de classe Triangle et de taille 3, l'afficher et récupérer le nombre d'étoiles retourné par affiche() dans une variable $n. Afficher la valeur de $n.
4. Créer un objet $tri6 de classe Triangle et de taille $_GET["taille"]. Tester la création et l'affichage de $tri6 en ajoutant dans l'URL de la page td1.php la chaîne « ?taille=6 ».

Bibliothèque de classes

Placer les classes HelloWorld, Tableau, Matrice et Triangle dans un fichier séparé nommé td1.class.php. Supprimer le code correspondant dans le fichier td1.php et y inclure à la place un fichier td1.class.php, avant les créations d'objets et les appels aux méthodes. Tester.

Validation HTML

Vérifier que le code HTML5 produit à l'aide de PHP (affichage du code source dans le navigateur) est valide en le copiant/collant dans https://validator.w3.org. Corriger les erreurs le cas échéant, jusqu'à ce que votre code soit valide.

Correction

```php
<?php // td1.class.php

     class HelloWorld {
          private $n = 10;
          public function afficherN() {
               echo "<p>Valeur de \$n = $this->n</p>\n";
          }
     }

     class Tableau {
          // private $t;
          protected $t;
          function __construct($param_tab) {
               $this->t = $param_tab;
          }
          public function afficheV() {
               echo "<ol>\n";
               foreach ($this->t as $valeur) echo "<li>$valeur</li>\n";
               echo "</ol>\n";
          }
          public function afficheKV() {
               echo "<ul>\n";
               foreach ($this->t as $cle => $valeur)
                    echo "<li>$cle : $valeur</li>\n";
               echo "</ul>\n";
          }
     }

     class Matrice extends Tableau {
          function __construct($param_tab) {
               parent::__construct($param_tab);
          }
          public function sommeMat() {
               $somme = 0;
               foreach ($this->t as $ligne)
                    foreach ($ligne as $valeur)
                         $somme += $valeur;
               return $somme;
          }
     }

     class Triangle {
          private $taille;
          function __construct($param_taille = 5) {
               $this->taille = $param_taille;
          }
          public function affiche() {
               $nb_etoiles = 0;
               if ($this->taille <= 20) {
                    if ($this->taille <= 0) $this->taille = 10;
                    $ligne_etoiles = "";
                    echo "<p>";
                    for ($i = 1; $i <= $this->taille; $i++) {
                         $ligne_etoiles .= "*";
                         $nb_etoiles += $i;
                         echo "$ligne_etoiles<br />\n";
                    }
                    echo "</p>\n";
               } else echo "<p>Triangle trop grand !</p>\n";
               return $nb_etoiles;
          }
     }
?>
```

```
<!DOCTYPE html> <!-- td1.php -->

<html lang="fr">

<head>
     <meta charset="utf-8" />
     <title>Programmation web - TD 1 exemple</title>
     <meta name="Author" content="Jérôme Darmont" />
     <meta name="Keywords" content="Web,PHP objet" />
</head>

<body>

<?php
     // Appel au fichier des classes
     require("td1.class.php");

     // Hello world!
     $h = new HelloWorld;
     $h->afficherN();
     // echo $h->n; ne fonctionne pas car l'attribut $n est privé

     // Tableaux et boucles (via les méthodes)
     $tscal = new Tableau(array("TF1", "France 2", "France 3"));
     $tscal->afficheV();
     $tassoc = new Tableau(array("Lundi" => "Purée",
          "Mardi" => "Patates sautées", "Mercredi" => "Frites",
          "Jeudi" => "Gratin dauphinois", "Vendredi" => "Hachis Parmentier"));
     $tassoc->afficheKV();
     $mat = new Matrice(array(   array(1, 3, 5),
                                 array(9, 0, 2) ) );
     echo "<p>Somme = " . $mat->sommeMat() . "</p>\n";

     // Boucles et tests (via les méthodes)
     $tri1 = new Triangle(5);
     $tri1->affiche();
     $tri2 = new Triangle(50);
     $tri2->affiche();
     $tri3 = new Triangle(-5);
     $tri3->affiche();
     $tri4 = new Triangle();
     $tri4->affiche();
     $tri5 = new Triangle(3);
     $n = $tri5->affiche();
     echo "<p>$n étoiles</p>\n";
     $tri6 = new Triangle($_GET["taille"]);
     $tri6->affiche();
?>

<p><a href="http://validator.w3.org/check?uri=referer">Validation HTML5</a></p>

</body>
</html>
```