Université Grenoble Alpes - IUT2 - Département informatique

## R3.01 - Programmation WEB - 2024

## Objectifs du TP:

- Utilisation de la structure d'une classe. Manipulation de données distantes.
- Lecture et extraction des données d'un fichier. Interaction simple avec mémoire.

## 1. Modèle Objet

Classe

Le langage PHP est un langage objet qui s'inspire fortement du langage Java. Le langage Java est lui même inspiré du langage C++. Ce qui suit présente l'orientation objet de PHP en comparaison d'autres langages.

TP 4 : Jukebox avec un fichier de données

PHP possède la notion de classe comme les autres langages. Javascript est particulier, car il existe une syntaxe qui semble définir des classes, alors que de manière stricte la notion de classe n'est pas présente en Javascript. C'est par contre le seul langage qui est tout objet, alors que dans les autres langages, seul les instances des classes sont des objets (i.e. pas les classes). Seul le langage C++ est compilé en langage machine et il est donc directement exécutable par le processeur. Les autres langages sont soit interprétés lors de l'exécution, soit semi compilés

dans un code intermédiaire : cela nécessite encore un programme pour l'exécution du code. Un code semi-compilé permet de vérifier une seul fois la syntaxe avant l'exécution. L'éxecution d'un code semi-compilé est aussi plus rapide qu'un code interprété (mais moins rapide que compilé). PHP dans une serveur comme Apache, possède un interpréteur, mais il peut être transformé en code intermédiaire comme en Java, lors de sa première exécution. La deuxième exécution du même script est alors plus rapide. Le typage statique fixe un type unique et non modifiable aux variables et aux paramètres. Le type de la valeur (ou de l'objet) doit correspondre au type de la variable. Le typage statique permet

la compilation car il faut connaitre à l'avance la taille (en octets) des données à stoker dans une variable. A l'inverse un typage dynamique ne fixe pas de type aux variables. Seules les valeurs ou les objets on un type. N'importe quel langage de programmation peut être utilisé coté serveur. Par contre, coté client dans la plupart des navigateurs seul le Javascript (et le Web Assembly) peut s'exécuter. Il est possible d'ajouter un interpréteur à certains navigateurs pour exécuter autre chose (comme du Java), mais cette possibilité a disparue des navigateurs les plus utilisés pour des raisons

économiques et de sécurité. **Javascript** Java

Oui et Non

C++

class Personne {

string nom;

int age;

public:

```
Objet
                                                                                              Tout est objet
                          Instances
                                                    Instances
                                                                   Instances
 Exécution
                          Interprété / Semi compilé
                                                                   Compilé
                                                                                              Interprété
                                                   Semi compilé
                                                   Statique
                          Dynamique
                                                                   Statique
                                                                                              Dynamique
 Typage
 Exécution coté serveur
                                                                                              Oui
                          Oui
                                                   Oui
                                                                   Oui
 Exécution coté client
                          Non
                                                   Non
                                                                   Non (Oui : WebAssembly)
                                                                                              Oui
Compaison de la syntaxe des classes dans 3 langages :
                   PHP
                                                                        Java
```

Oui

Oui

private int \$age;

class Personne {

private string \$nom;

Oui

```
Personne(String n, int a) {
                       function __construct(string $n, int $a) {
                                                                                                                                Personne(string n, int a) {
                         this->nom = t;
                                                                                this.nom = n;
                                                                                                                                  this->nom = n;
                         $this->age = $a;
                                                                                                                                  this->age = a;
                                                                                this.age = a;
Classe
                       function show() : string {
                                                                                                                                string show() {
                                                                              String show() {
                         $out = "Bonjour, je suis ";
                                                                                                                                  string out = "Bonjour, je suis ";
                                                                                String out = "Bonjour, je suis ";
                         $out .= $this->nom;
                                                                                out += this.nom;
                                                                                                                                  out += this->nom;
                                                                                out += ". J'ai ".this.age." ans.";
                                                                                                                                  out += ". J'ai "+this->age+" ans.";
                         $out .= ". J'ai ".$this->age." ans.";
                         return $out;
                                                                                return out;
                                                                                                                                  return out;
                    $p = new Personne("Julien",32);
                                                                           Personne p = new Personne("Julien",32);
                                                                                                                              Personne* p = new Personne("Julien",32);
                    $message = $p->show();
                                                                           String message = p.show();
                                                                                                                              string message = p->show();

    this: est optionnel en C++ et Java, mais ils est obligatoire en PHP (avec un '$').

 • $ : en PHP, le caractère dollar est obligatoire sur toutes les variables et les attributs SAUF pour l'accès à un attribut. Mettre un dollar sur le nom d'un attribut est possible mais est une
   indirection : le nom de l'attribut est dans la variable.
 • Allocation mémoire: dans les 3 langages, l'objet est créé sur le tas, avec un accès global par adresse (référence ou handle). Sa durée de vie est celle du programme. Seul C++ permet de
   détruire explicitement un objet. Seul C++ permet de créer des objets à portée locale sur la pile. PHP et Java récupère la mémoire perdue par des objets inaccessibles avec un ramasse
   miette.
 • Visibilité: par défaut, la visibilité en PHP est public, c'est à dire visible par tous. En Java, par défaut, la visibilité est limité au package. En C++ la visibilité par défaut est private.
```

class Personne {

private String nom;

private int age;

dans les fonctions qui utilisent ce tableau. Le modèle objet permet d'encapsuler des données avec des méthodes. Cet exercice vous propose donc de placer les définitions des abréviations dans une classe, et de transformer les fonctions de transformation des abréviations en méthodes. Nous ajoutons également une méthode qui replace de manière automatique toute abréviation connue.

<abbr title="description de l'abréviation">abréviation</abbr>

transforme un texte en montrant avec

'API' => 'Application Programming Interface',

'WYSIWYG' => 'What You See Is What You Get',

'XML' => 'eXtensible Markup Language'

private \$abbr = array();

// Pour remplacer des abbréviations dans un texte de manière automatique class Abbr { // La table de toutes les abréviations

```
// Construit en HTML l'abréviation $abbr ou ressort simplement $abbr
  // si $abbr n'est pas une abréviation connue
  function show(string $abbr) : string {
    /* A COMPLETER */
    return $abbr;
  // Filtre un texte pour inserer les abréviations trouvées
  function filter(string $text): string {
    $out = '';
    /* A COMPLETER */
    return $out;
  // Donne le nombre d'abréviations connues
  function size() : int {
    return count($this->abbr);
// L'objet abréviation pour le domaine informatique
$abbr = new Abbr($abbr_computing);
?>
<!DOCTYPE html>
<html lang=fr dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Protocole HTTP</title>
  </head>
  <body>
    <h1>Protocole HTTP</h1>
    >
      <?= $abbr->filter('
      Le protocole HTTP permet au navigateur de demander des données à un serveur.
      Ces données sont principalement au format HTML codées
      en ASCII, mais cela peut être aussi des images en JPEG ou des vidéo en MPEG par exemple.
      Pour indiquer au client quel type de données il reçoit,
      cette information est codée en MIME dans l\'entête de la réponse.
      ') ?>
    Il y a <?= $abbr->size() ?> abréviations connues.
  </body>
</html>
• Téléchargez le fichier zip qui contient le code incomplet : data/abbr.zip . Ce code ne change aucune abréviation et il affiche qu'il y a 0 abréviations.

    Complétez le constructeur pour qu'il initialise l'attribut avec son paramètre. On doit alors voir s'afficher qu'il y a 38 abréviations.

    Réalisez la méthode

     show
  en utilisant l'attribut de la classe.

    Réalisez la méthode

     filter
  en utilisant la methode
     show
```

## n'a pas été transformée pourquoi? Pour solutionner ce problème, vous pouvez utiliser preg split à la place de

1W

suivante:

ASCII

explode

sur chaque mot du texte.

Améliorations : Bonus

Notez que l'abréviation

. Vous devrez alors choisir les options (flags) PREG\_SPLIT\_NO\_EMPTY | PREG\_SPLIT\_DELIM\_CAPTURE

Nous voulons maintenant que la page principale se construise automatiquement en fonction des données présentes dans le fichier jukeboxData.txt. Ce fichier est structuré de la manière

• fgets: lecture d'une ligne, jusqu'à la marque de fin de ligne ou la fin du fichier. Attention : la marque de fin de ligne est lue également ("\n"). On peut la supprimer grâce à la fonction rtrim

Cette fonction prend en entrée le nom d'un fichier au format texte, où chaque ligne est une suite de champs délimités par le motif \$delimiter, qui a une valeur par défaut (le caractère | ).

Chaque ligne est découpée dans un tableau. Chaque tableau est assemblé dans un tableau qui finalement est rendu en résultat de cette fonction. Note : si le délimiteur est une virgule, et

o Vous pouvez transformer en une seul opération un texte en tableau de mots à l'aide de la fonction explode.

pour que soient retournés tous les caractères de séparation. Dans l'expression régulière (pattern), vous pouvez utiliser la classe

2. Site Web: un Jukebox dynamique Cet exercice est la suite du site WEB du TP précédent. Dans cette partie, le site WEB va devenir dynamique : les musiques à afficher et à jouer ne seront plus "en dur" dans le fichier HTML, mais dans un fichier que le php devra parcourir pour afficher la liste des musiques disponibles.

Dads Groin Twerk

array(9) {

array(2) {

[0]=>

[1]=>

array(2) {

[0]=>

[1]=>

[1]=>

string(4) "Dads"

string(11) "Groin Twerk"

string(19) "The Black Eyed Peas"

string(21) "Just Can't Get Enough"

[0]=>

[1]=>

2.1 Affichage dynamique

Chaque ligne décrit une musique.

Exemple du contenu du fichier dynamicJukebox.txt:

PREG\_SPLIT\_DELIM\_CAPTURE

The Black Eyed Peas I Gotta Feeling The Black Eyed Peas Just Can't Get Enough Adhitia Sofyan After The Rain Nous avons besoin d'une fonction PHP readJukeboxData qui va analyser ce fichier texte et va produire le résultat de son analyse dans un tableau de tableau. Chaque entrée du premier tableau est une musique, décrite par un tableau contenant le nom du groupe puis le nom de la musique. Par exemple, le début de cette structure est :

• Une musique est composée du nom du groupe suivi du nom de la musique séparée par le caractère |.

qui représente tous les caractères qui ne sont pas les lettres d'un mot. Attention, l'option

necessite d'usage de parenthèses pour délimiter et récupérer correctement les motifs de séparation.

string(15) "I Gotta Feeling" [2]=> array(2) { [0]=> string(19) "The Black Eyed Peas"

fopen: ouvre un fichier à partir du chemin sur le disque.

 feof: vrai si la dernière opération de lecture du flux a rencontré la fin de fichier (ex: fgets). Rappel : le test de fin de fichier foef est vrai uniquement si la dernière lecture n'a pas pu se réaliser car il n'y a plus rien à lire. Il faut donc réaliser une première lecture (ex: fgets) avant de rentrer dans la boucle. La boucle à écrire est de type TRAITER / AVANCER, cela veut dire qu'elle doit se terminer par une lecture. Travail à réaliser : A partir de cet exercice, nous developpons du code PHP avec des tests associés. Le code ne doit pas être intégré dans du HTML tant que les tests ne sont pas passés avec succès. 1. Récupérer les données depuis : data/dynamicJukebox.zip

2. Dans le fichier readJukeboxData.php réaliser la fonction :

3. Tester cette fonction hors du navigateur dans un terminal shell :

php test\_readDelimitedData.php

array(2) {

string(4) "Dads"

string(11) "Groin Twerk"

string(19) "The Black Eyed Peas"

string(19) "The Black Eyed Peas"

string(21) "Just Can't Get Enough"

string(15) "I Gotta Feeling"

[0]=>

[1]=>

array(2) {

[0]=>

[1]=>

array(2) {

[0]=>

[1]=>

array(2) {

[0]=>

[1]=>

array(2) {

[0]=>

[5]=>

[1]=>

[2]=>

[3]=>

Vous devez obtenir le résultat suivant : array(9) { [0]=>

La lecture d'un fichier se programme en PHP de manière similaire en langage C. Voici les fonctions à utiliser pour cet exercice :

function readDelimitedData(string \$filename, string \$delimiter='|') : array

si la première ligne est le nom des champs, alors ce format de fichier s'appelle le format CVS.

• <a href="rtrim">rtrim</a>: produit une nouvelle chaîne de caractères sans les caractères d'espace ou de fin de ligne en bout.

• explode: découpe une chaîne de caractères sur un motif de caractères. Place le résultat dans un tableau.

```
[1]=>
  string(14) "After The Rain"
[4]=>
array(2) {
  [0]=>
```

string(14) "Adhitia Sofyan"

string(5) "Ke\$ha"

string(6) "Tiktok"

```
string(7) "Justice"
  [1]=>
  string(7) "Genesis"
[6]=>
array(2) {
  [0]=>
  string(7) "Justice"
  [1]=>
  string(9) "D.A.N.C.E"
[7]=>
array(2) {
  [0]=>
  string(9) "Passenger"
  [1]=>
  string(8) "Facebook"
[8]=>
array(2) {
  [0]=>
  string(9) "Passenger"
  [1]=>
  string(16) "Community Centre"
```

navigateurs) ne renseignent pas ce champ.

Travail à réaliser :

2.2 Construction du fichier des musiques Bonus

 opendir: qui ouvre un répertoire. • readdir: qui lit le contenu d'un répertoire ouvert. • pathinfo: qui détaille le contenu d'un chemin vers un fichier. Cela permet par exemple d'extraire l'extension du nom du fichier. is dir: qui indique si le fichier est un répertoire.

Le parcours d'un répertoire est similaire à un parcours de fichiers, mais utilise les fonctions suivantes :

Limitez au maximum le PHP : laissez le HTML même à l'interieur de la boucle foreach qui parcourt le tableau.

o Utilisez la balise d'échappement echo courte exactement à l'endroit où le PHP doit insérer du texte dans le HTML.

Le fichier des musiques jukeboxData.txt peut être produit automatiquement en parcourant les sous-répertoires et les fichiers du répertoire data.

staticJukebox.html. Voici des conseils pour construire cette page :

php createDataList.php > jukeboxData.txt

noté: et la fin de bloc est notée endforeach; à la place des {...}

• Réaliser le fichier createDataList.php qui s'utilise dans un terminal shell pour créer le fichier du contenu du jukebox. Il s'utilise de la manière suivante :

Sujet de TP Programmation WEB (PHP).

4. Vous êtes pret maintenant à utiliser cette fonction dans la page HTML dynamique dynamicJukebox.php construite avec PHP. Cette page est similaire à la page statique

• Le PHP en début de fichier vous permet d'inclure le fichier readDelimitedData.php et d'appeler la fonction readDelimitedData sur le fichier jukeboxData.txt.

o Utilisez la syntaxe alternative du foreach pour rendre le code plus lisible lorsque l'on encadre du HTML avec des parties de PHP. Dans cette syntaxe, le début du bloc de la boucle est

5. Finalement, modifiez le fichier playPath.php pour que le bouton revienne automatiquement à la page précédente, c'est-à-dire soit la page statique : staticJukebox.html, soit la page

dynamique: dynamicJukebox.php, en fonction de là d'où l'on vient. C'est possible simplement grâce au tableau super global SERVER. Ce tableau contient des informations sur le

6. Question bonus: trouver une solution pour que de la page playPath.php on revienne automatiquement à la bonne page précédente sans utiliser \$\_SERVER["HIIP\_REFERER"].

contexte d'activation du fichier PHP. L'entrée "HTTP\_REFERER" est intéressante car elle indique l'adresse de la page qui a conduit le client à la page courante. Attention, certains clients (i.e.

Instance Appel méthode 1.1 Abréviations dans une classe Vous avez fait précédement un exercice sur les abrévations. Dans cet exercice, vous avez dû déclarer un ensemble d'abréviations dans un tableau global. Vous avez dû utiliser la directive global Dans le code ci-dessous, les abréviations sont placées dans un tableau global comme dans l'exercice précédent. Ce tableau devra être passé au constructeur d'un objet de type Abbr et sera accessible à partir d'un attribut privé. La méthode show(\$abbr) transforme une abréviation connue \$abbr en une structure HTML . Si l'abréviation n'est pas connue, alors retourne le mot contenu dans le paramètre \$abbr sans le modifier. La méthode filter(\$text) show les abréviations qui on été reconnues dans le texte. La partie HTML montre un exemple d'utilisation. Notez l'attribut \$abbr a été initialisé à un tableau vide. Cela permet à la méthode size de fonctionner. Initialiser les variables et les attributs est bonne pratique. En PHP, cela permet d'indiquer le type associé à la variable. Même si en PHP la variables ne sont pas typées, c'est une bonne pratique de limiter les variables et les attributs à un seul type de valeur qui a toujours la même signification. <?php // Liste des abréviations du domaine informatique \$abbr\_computing = array( 'ADSL' => 'Asymetric Digital Subscriber Line', 'AJAX' => 'Asynchronous JavaScript and XML',