Exploration

écriture et lecture POO en PHP

Rédigé par

David ROUMANET
Professeur BTS SIO

Changement

Date	Révision

Sommaire

Nomenclature :

- **Assimiler** : cours pur. Explication théorique et détaillée (globalement supérieur à 4 pages).
- **Décoder** : fiche de cours, généralement inférieure à 5 pages.
- **Découvrir** : Travaux dirigés. Faisable sans matériel.
- **Explorer** : Travaux pratiques. Nécessite du matériel ou des logiciels.
- **Mission** : Projet encadré ou partie d'un projet.
- **Voyager** : Projet en autonomie totale. Environnement ouvert : Vous êtes le capitaine !

A Introduction

L'activité qui suit a pour seul but de montrer la corrélation entre des données dans une base, et l'usage d'une collection d'objet.

A.1 Préparation

Pour cela, nous allons simplement utiliser une table IMC contenant :

- Id (integer, clé primaire)
- nom (varchar[50])
- poids (decimal)
- taille (decimal)
- dateMesure (date)

A.2 Table mySQL

Lors de la création d'une base de données, privilégiez la création d'une base en UTF-8 et utilisant le moteur InnoDB qui respecte les contraintes d'intégrité (ici, dans cette activité, ce ne sera pas utile, mais prenez de bons réflexes)...

Voici le détail de la table IMC que nous utiliserons.

```
CREATE TABLE imc (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(255) NOT NULL,
    poids DECIMAL(10,2) NOT NULL,
    taille DECIMAL(10,2) NOT NULL,
    date DATE NOT NULL
);
```

Jeu de données

Pour tester nos données, j'ai utilisé une IA pour générer un jeu de données diversifié :

```
INSERT INTO imc (nom, poids, taille, date) VALUES
('Alice', 60.0, 1.65, '2023-03-01'),
('Bob', 75.0, 1.75, '2023-06-01'),
('Charlie', 85.0, 1.80, '2023-09-01'),
('Dave', 90.0, 1.70, '2023-12-01'),
('Eve', 55.0, 1.60, '2024-03-01'),
('Frank', 80.0, 1.85, '2024-06-01'),
('Grace', 65.0, 1.70, '2024-09-01'),
('Harry', 95.0, 1.90, '2024-12-01'),
('Ingrid', 70.0, 1.65, '2025-03-01'),
('Jack', 100.0, 1.80, '2025-06-01');
```

L'intérêt est de faire faire à l'IA un travail simple mais répétitif pour nous.

B Code PHP

B.1 Page d'affichage de la page

L'idée est d'utiliser une classe, dont le format est le même que les données dans la base.

B.1.1 Classe en PHP

Le code est le suivant :

IMC.php

```php
<?php
class IMC {
    private $id;
    private $nom;
    private $poids;
    private $taille;
    private $date;

    public function __construct($id, $nom, $poids, $taille, $date) {
        $this->id = $id;
        $this->nom = $nom;
        $this->poids = $poids;
        $this->taille = $taille;
        $this->date = $date;
    }

    public function getId() {
        return $this->id;
    }

    public function getNom() {
        return $this->nom;
    }

    public function getPoids() {
        return $this->poids;
    }

    public function getTaille() {
        return $this->taille;
    }

    public function getDate() {
        return $this->date;
    }

    public function calculerIMC() {
        return $this->poids / pow($this->taille, 2);
    }
}
?>
```

Comme dans la plupart des langages capables de gérer des classes et des objets, on retrouve la notion de portée (`private`, `public`), le `this` permettant d'adresser les attributs de l'objet courant et les méthodes pour accéder aux attributs privés.

En PHP, le constructeur ne porte pas le nom de la classe, mais est nommé `__construc()`.

B.1.2 Inclusion en PHP

Dans un fichier PHP, il est possible d'inclure un morceau de code en provenance d'un autre fichier. C'est un peu comme un module en ES6 (JavaScript).

Voici donc un exemple pour afficher la table en utilisant la classe IMC.php

afficherTable.php

```php
<?php
include('IMC.php'); // inclusion du fichier IMC.php

// Paramètres de connexion à la base de données
$serveur = "localhost";
$utilisateur = "votre_utilisateur";
$mot_de_passe = "votre_mot_de_passe";
$base_de_donnees = "votre_base_de_donnees";

// Fonction pour se connecter à la base de données et retourner un objet mysqli
function connecterBdd() {
    global $serveur, $utilisateur, $mot_de_passe, $base_de_donnees;
    $conn = new mysqli($serveur, $utilisateur, $mot_de_passe, $base_de_donnees);
    if ($conn->connect_error) {
        die("Connexion échouée : " . $conn->connect_error);
    }
    return $conn;
}

// Fonction pour récupérer les données de la table IMC sous forme de tableau d'objets IMC
function getImcs() {
    $conn = connecterBdd();
    $sql = "SELECT id, nom, poids, taille, date FROM imc";
    $resultat = $conn->query($sql);
    $imcs = array();
    if ($resultat->num_rows > 0) {
        while($ligne = $resultat->fetch_assoc()) {
            $imc = new IMC($ligne["id"], $ligne["nom"], $ligne["poids"], $ligne["taille"],
$ligne["date"]);
            $imcs[] = $imc;
        }
    } else {
        echo "Aucun résultat trouvé.";
    }
    $conn->close();
    return $imcs;
}

// Fonction pour afficher les données dans un tableau HTML
function afficherTableauImcs($imcs) {
    echo "<table border='1'>";
    echo "<tr><th>ID</th><th>Nom</th><th>Poids</th><th>Taille</th><th>Date</th><th>IMC</th></tr>";
    foreach ($imcs as $imc) {
        echo "<tr>";
        echo "<td>" . $imc->getId() . "</td>";
        echo "<td>" . $imc->getNom() . "</td>";
        echo "<td>" . $imc->getPoids() . "</td>";
        echo "<td>" . $imc->getTaille() . "</td>";
        echo "<td>" . $imc->getDate() . "</td>";
        echo "<td>" . $imc->calculerIMC() . "</td>";
        echo "</tr>";
    }
    echo "</table>";
}

// Appel des fonctions pour récupérer et afficher les données
$imcs = getImcs();
afficherTableauImcs($imcs);
?>
```

Le code précédent récupère les lignes contenues dans la table IMC et l'insère à l'aide d'une boucle dans un tableau $imc qui contient les objets créés.

```
while($ligne = $resultat->fetch_assoc()) {
     $imc = new IMC($ligne["id"], $ligne["nom"], $ligne["poids"], $ligne["taille"], $ligne["date"]);
     $imcs[] = $imc;
     return $imcs;
}
```

L'intérêt est faible, car nous n'effectuons pas de traitement sur les différents objets pour le moment, mais nous pourrions travailler sur les objets sans avoir besoin de relancer une requête vers la base de données (en termes de latence, l'accès réseau à la base est rapide sur un LAN, mais peut devenir très long sur un téléphone ayant une mauvaise connexion).

Exemple d'utilisation

Nous pouvons construire une fonction qui fasse le calcul de la moyenne des IMC et en l'insérant dans le fichier AfficherTable.php :

```
function calculerMoyenneImc(array $imcs) {
    $sommeImc = 0;
    $nbImcs = count($imcs);
    foreach ($imcs as $imc) {
        $sommeImc += $imc->calculerIMC();
    }
    return $sommeImc / $nbImcs;
}
```

et à la fin du fichier (juste après l'appel de la fonction AfficherTableauImc()

```
afficherTableauImcs($imcs);
$moyenneImc = calculerMoyenneImc($imcs);
echo "La moyenne des IMC est de : " . $moyenneImc;
```

B.2 Résultat

Le résultat devrait être le suivant :

ID	Nom	Poids	Taille	Date	IMC
1	Alice	60.00	1.65	2023-03-01	22.038567493113
2	Bob	75.00	1.75	2023-06-01	24.489795918367
3	Charlie	85.00	1.80	2023-09-01	26.234567901235
4	Dave	90.00	1.70	2023-12-01	31.141868512111
5	Eve	55.00	1.60	2024-03-01	21.484375
6	Frank	80.00	1.85	2024-06-01	23.374726077429
7	Grace	65.00	1.70	2024-09-01	22.491349480969
8	Harry	95.00	1.90	2024-12-01	26.315789473684
9	Ingrid	70.00	1.65	2025-03-01	25.711662075298
10	Jack	100.00	1.80	2025-06-01	30.864197530864

La moyenne des IMC est de : 25.414689946307

Il est à noter que l'utilisation d'une collection d'objet est un encombrement de la mémoire vive : ici, nous n'avons que peu de données, mais les traitements sur une grande base de données ne se ferait pas de cette manière.