

Examen PHP POO

Les réponses aux questions doivent être rédigées avec soin et placées dans un fichier doc. On écrira son code PHP dans des fichiers séparés. On placera l'ensemble de l'examen dans un dossier portant son nom.

Bonne rédaction.

1) Exercice (3 points)

On Ajoutera à la classe ci-dessous une méthode qui permettra d'afficher le produit des nombres \$_p et \$_q.

```
<?php
Class Calcul{
    private $_p = 11 ;
    private $_q = 23 ;
}
?>
```

2) Exercice (3 points)

Sachant que **throw** permet de rejeter une exception pour qu'elle soit traitée par une méthode de plus haut niveau, trouver une solution pour que le message : "J'aimerais être affiché !", s'affiche à l'écran.

```
<?php
try {
    try {
        throw new Exception("Un message");
    } catch (Exception $e) {
        echo $e->getMessage();
    }
} catch (Exception $e) {
    echo "J'aimerais être affiché !";
}
?>
```

3) Exercice (7 points)

On utilisera la couche PDO pour réaliser la classe **Connect**. On utilisera la méthode `__construct()` de PHP pour initialiser une connexion à la base de données. Les erreurs seront traitées à l'aide de l'objet Exception de PHP.

On créera également deux méthodes permettant respectivement de récupérer la connexion à la base de données et de fermer celle-ci.

4) Exercice (5 points)

On créera la classe Factory, une classe possédant une méthode statique permettant d'instancier l'objet User sans instancier la classe Factory. La classe User possède une méthode publique qui affichera le nom d'un utilisateur.

5) Exercice (7 points)

En cours, on a vu comment utiliser la méthode magique `__call()` dans la classe Model, pour effectuer des SELEC sur une table.

On créera, à l'aide de la méthode magique `__call()`, la possibilité d'ajouter des valeurs à une table. On proposera une API du type « `insertInTable($value)` ».

6) Exercice (5 points)

On demande de décrire le fonctionnement du motif Singleton dans le détails et de donner un exemple d'utilisation de ce motif. On donne un exemple de singleton ci-dessous :

```
<?php
class Singleton {

    // à expliquer
    private static $_instance = null;

    // à expliquer
    private function __construct() {}

    // à expliquer
    public static function getInstance() {
        if (is_null(self::$_instance))
            self::$_instance = new self();
        return self::$_instance;
    }

    // à expliquer
    private function __clone() {}
}
?>
```

7) Exercice (vocabulaires) (5 points)

Qu'appelle-t-on une liaison pour les classes ?

Pour l'héritage qu'appelle-t-on le principe « est un » ?

Quelle est la liaison la plus forte entre deux classes ?

Lorsqu'on définit des classes quel principe doit-on tout le temps appliquer pour rendre les choses modulables, il faut penser qu'une classe doit pouvoir être utilisée comme des éléments autonomes logiques ?

On donnera une définition du couplage applicatif et on essayera de donner son intérêt dans les applications conséquentes.