

## TP Programmation Objet avec PHP

Objectif : construire une application de gestion des données d'une table Eleve à l'aide de la POO et composants PHP.

### La table : version simple

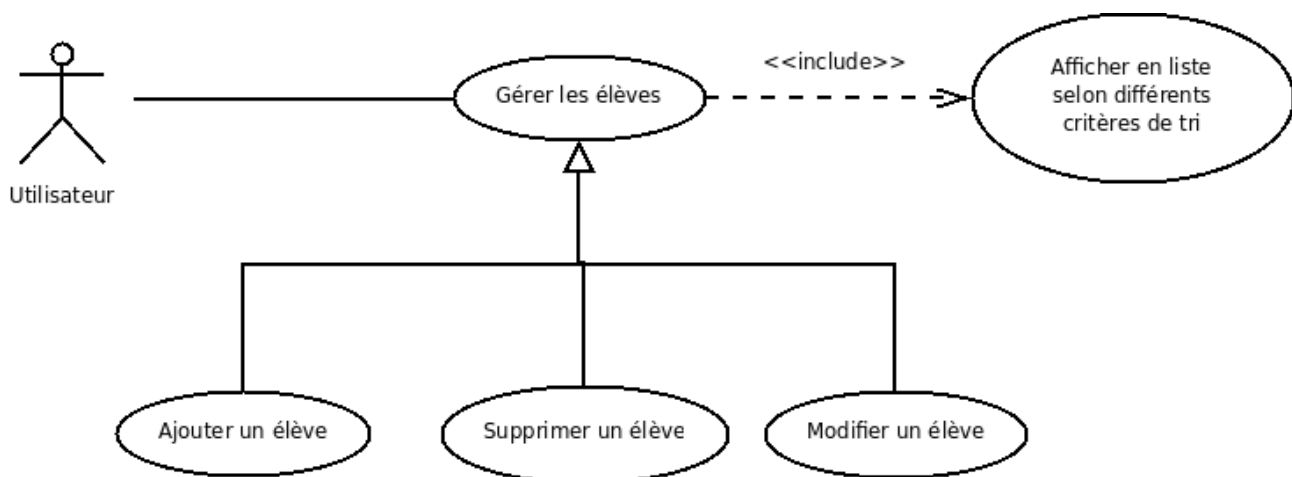
Eleve	id	nom	prenom	DateNaissance	internet

### Les fonctionnalités attendues

L'application proposera à l'utilisateur de

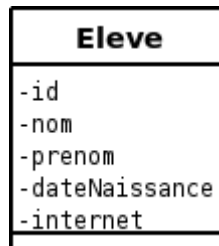
- Lister les élèves selon différents critères de tri
- Ajouter un élève
- Modifier un élève
- Supprimer un élève (après confirmation)

Voici le modèle des cas d'utilisation de l'application :



## Contraintes techniques

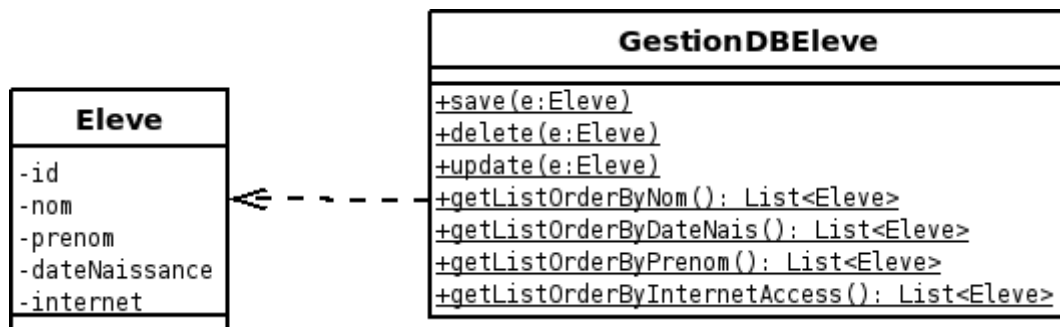
Représentation d'un élève (ligne de la table Eleve) en mémoire se fera sur la base d'une classe.



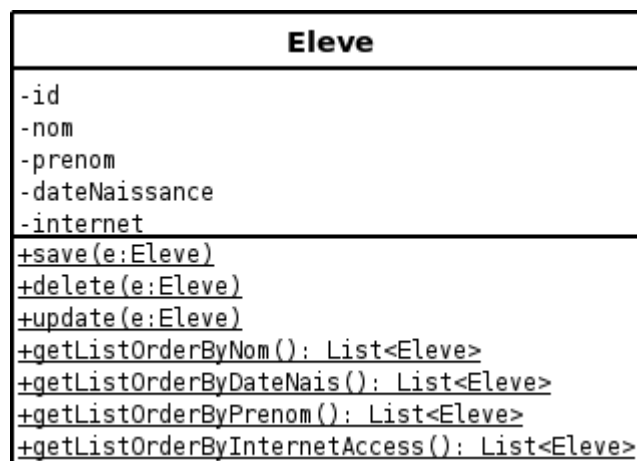
Prévoir un **constructeur** et les **accesseurs**.

Typiquement il y a deux façons d'opérer la gestion des la persistance (stockage dans une base de données)

- Soit une classe se charge du travail (CRUD), ici GestionDBEleve (les méthodes peuvent être de classe ou d'instance)



- Soit les opérations CRUD sont représentées par des méthodes de classe (**static**) de la classe Eleve.



Dans les deux cas, une autre classe nommée **SingletonPDO** sera chargée de fournir une instance de PDO, conformément à la présentation qui vous a été faite. Pour éviter d'instancier plusieurs fois une connexion à la base de données, la classe **SingletonPDO** sera implémentée selon le modèle de conception connu sous le nom de *singleton*. Une entrée ici :

<http://www.apprendre-php.com/tutoriels/tutoriel-47-classe-singleton-d-acces-aux-sgbd-integrant-pdo.html>

## Exemple de code pour bien démarrer

Exemple d'utilisation (index.php) : Ce sont des objets de type Eleve qui sont directement réceptionnés. Cela peut s'avérer très pratique lorsque les objets sont munis de responsabilité, comme ici un algorithme simpliste de détermination du login.

```
<?php require_once('GestionDBEleve.php'); ?>
<html>
<head><title>Test PDO</title> </head>
<body>
<?php
    /** obtient une entrée vers le résultat de la requête, côté SGBDR **/
    $stmt = GestionDBEleve::getListOrderByNom();

    /** demande le résultat sous la forme d'une collection d'objets de type Eleve **/
    $eleves = $stmt->fetchAll(PDO::FETCH_CLASS, 'Eleve');

    /** exploite un à un les objets de la collection **/
    /** pour la démonstration, on a
        un accès à la valeur d'un attribut privé par son accesseur (nom)
        un accès à un attribut public (prenom)
        un accès à une méthode (getLogin)
    **/
    foreach ($eleves as $eleve) {
        print $eleve->getNom() . ' : ' . $eleve->prenom . ' = ' . $eleve->getLogin() . '<br />';
    }
?>
</body>
</html>
```

Un version de la classe Eleve à compléter (constructeur, attribut etc.)

```
<?php
class Eleve {
    private $nom;
    public $prenom;

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

    public function getLogin() {
        return strtolower($this->nom{0} . $this->prenom);
    }
}
```

Exemple de classe GestionDBEleve :

```
<?php
require_once('SingletonPDO.php');
require_once('Eleve.php');

abstract class GestionDBEleve {
    public static function getListOrderByNom() {
        $pdo = SingletonPDO::getInstancePDO();
        /* return a PDOStatement object */
        return $pdo->query("SELECT * FROM eleve ORDER BY nom");
    }
}
```

Une version simplifiée du singleton :

```
<?php
class SingletonPDO {
    static private $instance = null;

    private $instancePDO = null;

    private function __construct() {
        try {
            $this->instancePDO = new PDO('mysql:host=localhost;dbname=test', 'root', '');
            $this->instancePDO->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);
        } catch (PDOException $e) {
            echo 'Erreur : ' . $e->getMessage();
            exit;
        }
    }

    public static function getInstancePDO() {
        if(is null(self::$instance)) {
            self::$instance = new SingletonPDO();
        }
        return self::$instance->instancePDO;
    }
}
```

Référence PDO sur le net :

- <http://www.php.net/manual/fr/book.pdo.php>
- <http://www.phppro.org/tutorials/Introduction-to-PHP-PDO.html>