

## TP5 – Travail de généricité sur les contrôleurs

Dans le TP précédent, un travail important a été fait sur la généricité des vues et des modèles. Le code a été nettement factorisé, et il ne reste des modèles que l'essentiel :

- Les attributs de classe,
- Les attributs d'instance,
- Une méthode d'instance (afficher).

Il nous reste à accomplir la même simplification au niveau des sept contrôleurs. Ce qui sera curieux, c'est que les classes contrôleurs sont purement statiques, et ne sont pas pour le moment vouées à être instanciées. Donc, si on regarde ce qu'il reste des modèles, tout ce qui concerne les instances de classes n'apparaîtra même plus dans les contrôleurs simplifiés.

Et donc, ces contrôleurs simplifiés se limiteront au strict minimum...

### Exercice 1 – Mise en place de la classe mère des contrôleurs

Dupliquez votre TP4/ex9 en TP5/ex1.

En principe, tous vos fichiers contrôleur sont actuellement très semblables :

```
1 <?php
2     require_once("modele/auteur.php");
3
4     class ControleurAuteur {
5
6 >         public static function lireAuteurs() {=
22
23 >         public static function lireAuteur() {=
32
33     }
34
35     ?>
```

L'idée est, comme pour les fichiers modèle, de créer un fichier de code commun.

Pour les fichiers modèles, nous avons créé un fichier `objet.php` qui définissait la classe `Objet`, dont héritaient toutes les autres classes de type `Modèle`.

Pour les fichiers contrôleurs, nous allons de la même manière créer un fichier `controleurObjet.php`, qui définira une classe `ControleurObjet`, dont hériteront toutes les autres classes de type `Contrôleur`.

1. Créez dans le répertoire TP5/ex1/controleur un fichier `controleurObjet.php`. Dans ce fichier, créez une classe `ControleurObjet`. Pour le moment, le code de cette classe est vide.
2. Faites en sorte que les sept classes contrôleurs héritent de cette nouvelle classe.
3. N'oubliez pas d'insérer cette nouvelle classe dans le routeur avec toutes les autres (dans le bon ordre, c'est mieux).

## Exercice 2 – Factorisation de la méthode `lireAuteurs` de la classe `Auteur`

Dupliquez TP5/ex1 en TP5/ex2.

Pour le moment, nous ne modifions que `ControleurAuteur` et `ControleurObjet`.

1. Créez, dans la classe `ControleurObjet`, une méthode

```
class ControleurObjet {  
  
    public static function lireObjets() {
```

qui sera amenée à remplacer toutes les autres méthodes `lireAuteurs`, `lireLivres`, ...  
Pour le moment, le code de votre méthode est vide.

2. Les actions `lireAuteurs`, `lireLivres`, ... définissent une variable `$titre` qui est utilisée pour afficher le titre de la page dans l'onglet du navigateur.

Définissez un attribut de classe `protected static $objet = "Auteur";` au niveau de la classe `ControleurAuteur`.

3. Pour commencer le code de `lireObjets`, récupérez dans une variable `$titre` la valeur de l'attribut `$objet` de la classe fille. Ensuite, grâce à cette variable, fabriquez le titre de la page. Vous vous arrangerez pour que :
  - « Auteur » donne « les auteurs »
  - « Livre » donne « les livres »
  - « Adherent » donne « les adherents » (nous ne tiendrons pas compte des accents)
  - ...

Vous pourrez utiliser la fonction `strtolower` de PHP.

4. Dans le menu, modifiez le lien vers l'action `lireAuteurs` en la remplaçant par l'action `lireObjets`.

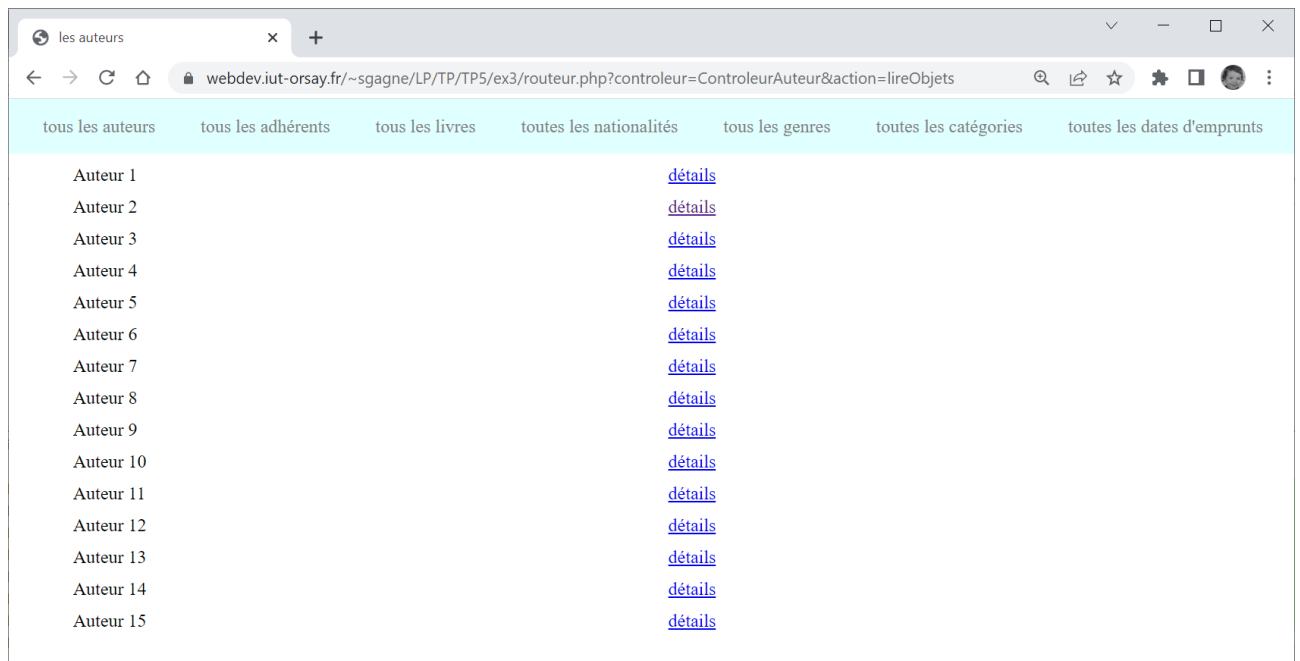
5. Pour tester votre mini-code, vous ne ferez pas faire d'autre calcul au contrôleur, puis vous insérerez dans l'ordres les vues `debut.php`, `menu.php` et `fin.php`. Vérifiez que le titre dans l'onglet des auteurs est correct.

### Exercice 3 – Factorisation de la méthode `lireAuteurs` de la classe `Auteur`, suite

Dupliquez TP5/ex2 en TP5/ex3.

1. Toujours dans le code de `lireObjets`, récupérez maintenant dans une variable `$tableau` les objets correspondant à la valeur de l'attribut `static $objet` de la classe fille.
2. Créez ensuite, comme dans `lireAuteurs`, le tableau `$tableauAffichage` qui va contenir toutes les lignes d'affichage.
3. Créez dans le contrôleur fille un autre attribut `static protected $cle = "..."` ; comme pour les classes modèle. Récupérez cet attribut dans la méthode `lireObjets`.
4. Remplissez le tableau d'affichage manière à afficher seulement :
  - a. le type de l'objet (Auteur, Livre, ...),
  - b. la valeur de son identifiant,
  - c. Le lien détails.

Voici un aperçu du futur rendu



5. Insérez la vue `lesObjets.php` au bon endroit. Faites Vérifiez que tout fonctionne !

## Exercice 4 – Factorisation des méthodes analogues à lireAuteurs des autres classes

Dupliquez TP5/ex3 en TP5/ex4 .

Dans chacune des autres classes contrôleurs filles, on va maintenant

- Supprimer les méthodes `lireAdherents`, `lireLivres`, ... dans les différentes classes contrôleurs.
- Ajoutez dans ces classes les bons attributs `protected static`, comme cela a été fait pour `ControleurAuteur`.

N'oubliez pas d'adapter le menu (voir ex2 question 4). Vérifiez que tout fonctionne !

## Exercice 5 – Factorisation des méthodes lireAuteur, lireLivre, ...

Dupliquez TP5/ex4 en TP5/ex5 .

En autonomie, mettez tout en place pour créer une méthode `lireObjet` dans la classe `ControleurObjet`, qui remplacera toutes les méthodes `lireAuteur`, `lireLivre`, ... des classes filles.

Faites en sorte que tout fonctionne !

Attention ! Les liens détails sont aussi à corriger pour que tout soit opérationnel...

A ce stade, si vous observez les contrôleurs, vous pouvez vous demander s'il est vraiment utile de les conserver, puisqu'ils ne contiennent plus que deux attributs static.

Pourtant, nous allons les conserver car à l'avenir il pourra y avoir une différence dans les futures méthodes qu'ils proposeront, et qui ne seront pas forcément générissables.