# Réponses TP POO

1. L’héritage :
   1. L’héritage est une technique qui permet de « donner » des attributs et des méthodes d’une classe parent a une ou plusieurs classes enfant.
   2. L’héritage est utilisé quand on a plusieurs classes qui possèdent les mêmes attributs/méthodes . Supposons qu’on a 2 classes **A** et **B** , toutes les 2 avec les attributs **1,2,3** et les méthodes **X()** et **Y().** Pour éviter la redondance, mais aussi pour une meilleure qualité de code, on définit une classe **C** qui possède les attributs et les méthodes cites plus haut, et on les fait hériter aux classes **A** et **B** ainsi , ces 2 classes n’ont pas d’attributs ces attributs ou méthodes définies dans leur corps, mais elles peuvent les utiliser.
2. L’interface :
   1. Une interface est une classe qui possède des signatures de méthodes, c’est-à-dire des définitions de méthodes sans implémentation. Une interface oblige les classes qui l’implémentent à créer toutes leurs méthodes .
   2. On les utilise quand on veut être sûr qu’une classe possède des méthodes particulières.

1. Classe Abstraite :
   1. Une classe abstraite est une classe qui ne va pas pouvoir être instanciée directement, c’est-à-dire qu’on ne va pas pouvoir manipuler directement. Elle permet de définir des attributs et des signatures de méthodes, mais pas de les instancier. Elle sera plus tard héritée par une autre classe.
   2. Une classe abstraite oblige donc les classes enfants à avoir les mêmes fonctionnalités, elle force l’héritage. Mais a la différence d’une interface, la classe abstraite peut posséder des signatures de méthodes mais aussi des attributs.
2. Propriété privée :
   1. En déclarant un attribut avec la propriété private, on interdit l’accès a cet attribut depuis l’extérieur. C’est-à-dire qu’on pourra modifier l’attribut que depuis l’intérieur de la classe qui l’a définie.
   2. On l’utilise quand on veut rendre notre propriété visible que dans la classe qui la définie.
3. Propriété protected :
   1. Les propriétés, méthodes définies avec le mot clef protected ne vont être accessibles que depuis l’intérieur de la classe qui les a définies ainsi que depuis les classes qui en héritent ou la classe parente.
   2. On utilise protected lorsqu’on souhaite rendre la propriété visible dans toutes les classes qui héritent de celle dans laquelle on l’a défini, y compris la classe parente. Mais pas a l’extérieur des classes.

**Ce projet permettra d’afficher les articles avec un CRUD**

**Ce projet permettra d’afficher les utilisateurs avec un CRUD**

**Il aura une interface d’administration permettant**

* **Le nombre d’articles**
* **Le nombre d’articles par utilisateur**
* **Les utilisateurs ayant le plus posté d’articles**

1. MCD : voir image DiagrammeBDD.png
2. Index->Routeur->contrôleur->model(entité) ->vue
3. ROUTAGE :

On va utiliser une classe classique avec une méthode statique (MAIN) : abstract public function main()