TP POO

Expliquez les concepts POO suivants en expliquant dans quel cas nous pouvons utiliser ces éléments et comment techniquement ils s’utilisent :

* L’héritage

L’heritage permet de définir une nouvelle classe à partir d'une classe existante, en héritant de toutes les caractéristiques (attributs et méthodes) de la classe d'origine

* L’interface
* La méthode abstraite

Une méthode abstraite est une méthode dont seule la signature (c’est-à-dire le nom et les paramètres) va pouvoir être déclarée mais pour laquelle on ne va pas pouvoir déclarer d’implémentation (c’est-à-dire le code dans la fonction ou ce que fait la fonction).

* La classe abstraite

Une classe abstraite c’est une classe qui va jamais s’instancier.

* Les propriété privée / protected

Le mot-clé public indique que les propriétés et méthodes d'un objet seront accessibles depuis n'importe où dans le programme

Le mot-clé private permet de déclarer des attributs et des méthodes qui ne seront visibles et accessibles directement que depuis l'intérieur même de la classe. C'est à dire qu'il devient impossible de lire ou d'écrire la valeur d'un attribut privé.

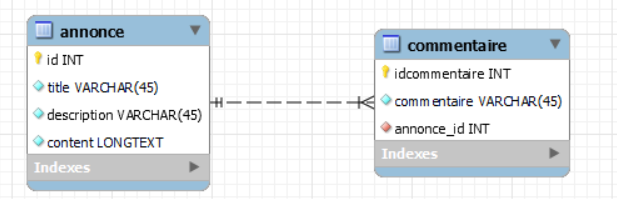
L'accès protégé ( protected ) est un intermédiaire entre l'accès publique et l'accès privé. Il permet d'utiliser des attributs et méthodes communs dans une classe parente et ses classes dérivées (héritantes).

* Les méthodes statiques

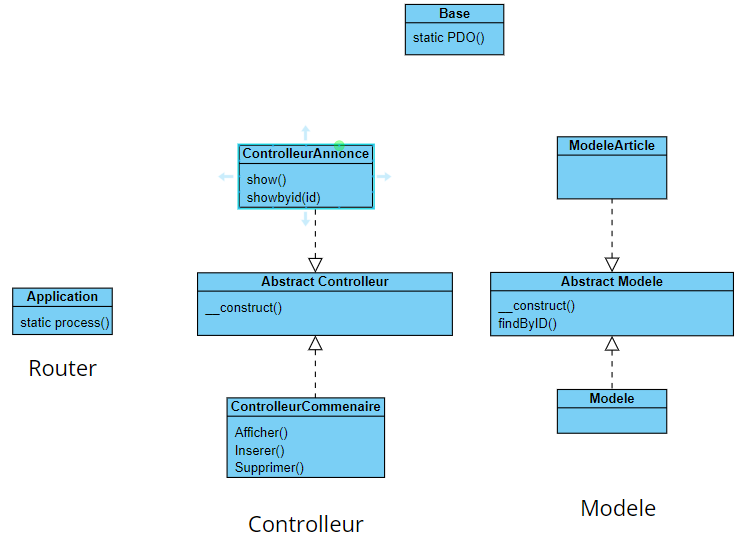
Une méthode static est une méthode qui n'agit pas sur des variables d'instance mais uniquement sur des variables de classe. Ces méthodes peuvent être utilisées sans instancier un objet de la classe. Les méthodes ainsi définies peuvent être appelées avec la notation classe. methode() au lieu de objet.

Reprenez le précédent TP PHP MVC d’annonces et réadapter le code sous forme de classe a l’aide des questions suivantes :

1. Décrivez le modèle conceptuel de donnée correspondant à la Base de donnée.



1. Décrivez la logique MVC intégrant le Routeur communiquant avec ce modèle
2. Conceptualisez les classes utilitaires
3. Conceptualisez les classes Modèles relié à cette problématique
4. Conceptualisez les classes Contrôleurs / vue relié à cette problématique



1. Codons, les vue, le routeur, les contrôleurs, les modèles