
TP6 : POO (Abstraction et encapsulation)

Ayoub KARINE (ayoub.karine@isen-ouest.yncrea.fr)

Consignes :

Pour chaque TP,

- créer un dossier TP_numTP (exemple : TP1)
 - dans le dossier TP_numTP, créer un fichier spécifique portant le nom du tp en cours suivi de votre nom (ex : <TP1-PHP-VotreNom.php)
 - dans l'en tête du fichier php, placer un titre indiquant le nom du TP entre une balise <h1>
 - Mettre une ligne horizontale (<hr>)
 - Mettre le numéro de l'exercice entre une balise <h2>
 - Mettre une ligne horizontale entre les les exercices
 - Pour chaque exercice, mettre le numéro de question entre une balise <h3>
-

Exercice 1

1. Créer une classe représentant une équipe de football ayant
 - a. nom et nombre de titres comme attributs publiques (ne pas utiliser de constructeurs)
 - b. une méthode `display()` affichant "L'équipe X a Y titres"
2. A l'extérieur de la classe,
 - a. Créer un objet de cette classe
 - b. Affecter des valeurs à ces deux attributs
 - c. Appeler la méthode `display()`
3. Rendre les attributs de la classe comme privés
4. Créer des getters et des setters pour accéder et pour modifier ces attributs
5. Refaire les questions 2.a, 2.b et 2.c
6. Modifier la classe en introduisant un constructeur
7. Refaire les questions 2.a, 2.b et 2.c
8. Créer un destructeur qui affiche "Destructor" à son appel
9. Créer
 - a. une constante contenant le texte : "Nombre d'équipe :"
 - b. un attribut et une méthode statiques permettant de retourner le nombre d'équipes (objets) créées. Appeler la constante de la classe dans la méthode statique

- c. Pour tester cela, créer à l'extérieur de la classe 4 équipes par exemple. L'affichage suivant sera généré :

Nombre d'équipe : 4

Vérifier aussi le nombre d'appel du constructeur

Exercice 2

1. Dans un fichier formulaire.php, créer une classe Formulaire représentant un formulaire HTML ayant
 - a. un constructeur qui crée le code d'en-tête du formulaire = l'élément <form>. Il a comme arguments le nom du fichier qui va récupérer les informations du formulaire et la méthode d'envoi des données
 - b. une méthode ajouterzonetexte() qui ajoute une zone de texte en HTML. Elle a comme arguments : texte (avant la zone du texte) et nom (valeur de name)
 - c. une méthode ajouterbouton() qui ajoute un bouton d'envoi en HTML. Elle a comme arguments : texte du bouton et nom (valeur de name)
 - d. une méthode getform() qui affiche tout le code HTML permettant la création du formulaire (texte + bouton)
2. Dans un fichier testformulaire.php,
 - a. appeler le fichier formulaire.php avec la fonction include

- b. créer un objet de la classe Formulaire et ajouter deux zones de texte et un bouton en utilisant les méthodes ajouterzonetexte() et ajouterbouton()
- c. afficher le formulaire

Votre nom :

Votre code :

Exercice 3

1. Créer, dans un fichier php, le formulaire HTML suivant :

Nom :

Prenom :

Mail :

Age :

Monsieur : ☐ Madame : ☐

2. Créer une classe permettant, via son constructeur, de récupérer les valeurs saisies dans le formulaire
3. Modifier la classe pour récupérer les valeurs saisies dans le formulaire via les getters
4. A l'extérieur de la classe, instancier un objet et afficher les valeurs saisies dans le formulaire en cas de clic sur le bouton "submit"