



W1 – Piscine PHP

Jour 10

Responsables Pédagogiques

pedagowac@epitech.eu

Bienvenue dans cette dixième journée de piscine Web@cadémie ! ☺

Au cours de cette journée nous allons approfondir la notion de programmation orientée-objet.

Faites un mail que vous enverrez à l'adresse donnée ci-dessus, dans lequel vous écrirez un poème qui sera destiné à votre pangolin préféré. L'heure limite de réception est fixée à dix heures, le jour même d'apparition du sujet.

Sommaire

Exercice 01 (2 pts)	3
Exercice 02 (2 pts)	4
Exercice 03 (2 pts)	5
Exercice 04 (2 pts)	6
Exercice 05 (2 pts)	7
Exercice 06 (2 pts)	8
Exercice 07 (2 pts)	9
Exercice 08 (2 pts)	10
Exercice 09 (2 pts)	11
Exercice 10 (2 pts)	12

Exercice 01 (2 pts)

Nom du rendu : Piscine_PHP_Jour_10/ex_01.php

Restrictions : Aucune

Créez une classe abstraite "Warrior". Elle aura une méthode publique nommée "attack" qui affichera "I'll kill you, poor noob !!!" suivi d'un retour à la ligne, et 2 méthodes publiques abstraites : "represent" et "shout".

Créez une classe "Booba" qui hérite de "Warrior". Sa méthode "represent" devra afficher "92" suivi d'un retour à ligne, et sa méthode "shout" devra afficher "Bah bien Morray !".

Créez une classe "LaFouine" qui hérite de "Warrior". Sa méthode "represent" devra afficher "78" suivi d'un retour à ligne, et sa méthode "shout" devra afficher "Je suis proprietaire !".

Exercice 02 (2 pts)

Nom du rendu : Piscine_PHP_Jour_10/ex_02.php

Restrictions : Aucune

Copiez le code de votre exercice 1. Vous allez apprendre les bonnes manières à "Booba".

Créez une interface "GoodManners" qui contiendra 3 méthodes publiques : "say_please", "say_thank" et "say_sorry", qui elle, prendra un paramètre nommé "name".

Elle contiendra aussi une constante nommée END_WORDS qui sera égale à ", dirais-je."

La classe "Booba" implémentera cette interface avec ses méthodes. Sa méthode "say_please" affichera "S'il-te-plait", sa méthode "say_thank" affichera "Merci", et sa méthode "say_sorry" prendra un nom en paramètre et affichera "Mille excuses " suivi du nom. Chacun de ces affichages se termineront par la constante END_WORDS suivi d'un retour à la ligne.

Exemple :

```
$booba = new Booba();
```

```
$booba->say_thank();
```

```
// affiche "Merci, dirais-je."
```

Exercice 03 (2 pts)

Nom du rendu : Piscine_PHP_Jour_10/ex_03.php

Restrictions :

- Utiliser le Trait avec le mot-clé "use".
- Utiliser la fonction `str_replace` 2 fois.
- Ne pas écrire 2 fois la fonction `sing_a_song`

Copiez le code de votre exercice 2. Vous allez apprendre à chanter à vos 2 warriors.

Créez un trait "Singer" qui contiendra une fonction "sing_a_song". Cette fonction affichera la chaîne de caractères "Trolololololol" suivi d'un retour à la ligne, puis transformera tous les 'o' en 'a' et réaffichera la chaîne suivi d'un retour à la ligne, pour enfin transformer tous les 'a' en 'i' et réaffichera la chaîne suivi d'un retour à la ligne.

Exercice 04 (2 pts)

Nom du rendu : Piscine_PHP_Jour_10/ex_04.php

Restrictions : Aucune

Créez une classe "Pony" qui aura 3 attributs publics : "gender", "name" et "color".

Faites en sorte qu'on puisse setter ces 3 attributs directement à la construction (dans le même ordre que cité précédemment).

Faites en sorte, qu'à la destruction du poney, la chaîne de caractères "I'm a dead pony." suivi d'un retour à la ligne soit affichée.

Faites aussi en sorte que lorsqu'on utilise la commande "echo" sur un "Pony", cela affiche "Don't worry, I'm a pony !" suivi d'un retour à la ligne.

Exercice 05 (2 pts)

Nom du rendu : Piscine_PHP_Jour_10/ex_05.php

Restrictions : Aucune

Copiez le code de votre exercice 4.

Ajoutez une méthode publique nommée "speak" qui affiche "Hiii hiii hiii" suivi d'un retour à la ligne.

Faites en sorte que lorsqu'on appelle une méthode qui n'existe pas dans la classe "Pony", cela affiche "I don't know what to do..." suivi d'un retour à la ligne.

Exercice 06 (2 pts)

Nom du rendu : Piscine_PHP_Jour_10/ex_06.php

Restrictions :

- Utiliser __get
- Utiliser __set

Copiez le code de votre exercice 5.

Mettez tous les attributs en privé, mais faites en sorte que l'on puisse quand même y accéder via get et set.

Quand on y accède en get, cela doit afficher "Ce n'est pas bien de getter un attribut qui est prive !" suivi d'un retour à la ligne, puis retourner l'attribut.

Quand on y accède en set, cela doit afficher "Ce n'est pas bien de setter un attribut qui est prive !" suivi d'un retour à la ligne, puis affecter l'attribut de la valeur.

Pour tout get ou set d'un attribut qui n'existe pas, cela doit afficher "Il n'y a pas d'attribut: " suivi du nom de l'attribut, d'un point et d'un retour à la ligne.

Exercice 07 (2 pts)

Nom du rendu : Piscine_PHP_Jour_10/ex_07.php

Restrictions : Aucune

Créez une classe "Body" qui aura dans le même ordre : un attribut private "head" qui vaudra 1 ; Et un attribut protected "arm", un attribut protected "leg", un attribut public "hand" et un attribut public "foot" qui vaudront tous 2.

Cette classe aura une fonction publique "print_inside_attributes" qui affichera dans un "foreach" tous les attributs à l'aide de parcours d'objets. L'affichage de chaque attribut se fera de la façon suivante : "attribut : valeur", suivi d'un retour à la ligne.

En dehors de cette classe, créez une fonction "print_object_attributes" qui prendra en paramètre un objet nommé "object", et qui affichera dans un "foreach" tous les attributs de cet objet à l'aide du parcours d'objets.

L'affichage de chaque attribut se fera de la façon suivante : "attribut => valeur", suivi d'un retour à la ligne.

Exercice 08 (2 pts)

Nom du rendu : Piscine_PHP_Jour_10/ex_08.php

Restrictions : Vous devrez utiliser le mot-clé "clone"

Créez une classe "Dolly" qui aura 3 attributs publics : "age", "animal" et "doctor".

Le constructeur prendra ces 3 attributs en paramètre (dans le même ordre que cité précédemment). Faites en sorte que lorsque qu'on clone un objet de cette classe, cela affiche "I will survive !" suivi d'un retour à la ligne.

En dehors de cette classe, créez une fonction "clone_object" qui prendra en paramètre un objet nommé "object" et qui retournera un clone de ce même objet.

Exercice 09 (2 pts)

Nom du rendu : Piscine_PHP_Jour_10/ex_09.php

Restrictions : Aucune

Créez une fonction "objects_comparison" qui prendra 2 objets en paramètre nommés respectivement "object1" et "object2".

Elle devra afficher "Objects are equal." suivi d'un retour à la ligne, si les 2 objets sont égaux ou "Objects are the same." suivi d'un retour à la ligne, si les 2 objets font référence à la même instance ou sinon dans les autres cas affichera un simple retour à la ligne

Exercice 10 (2 pts)

Nom du rendu : Piscine_PHP_Jour_10/ex_10.php

Restrictions : Aucune

Créez une classe "Pangolin" qui aura 2 attributs publics : "friends" de type array et "skills" de type Skill qui sera une classe définie par nos soins.

Le constructeur prendra ces 2 attributs en paramètre (dans le même ordre que cité précédemment).

Faites en sorte qu'il ne soit pas possible de passer un autre type que le type initial de chaque attribut.

De plus, on devra pouvoir passer le type "null" à l'argument qui correspond à l'attribut "friends".