

Exercice

Héritages

Consignes de l'exercice

- Sur index.php, remplacer "Votre nom" dans la méthode \$humain->setNom() par votre prénom.
- Ajouter à la classe personnage une propriété protégée \$vie
- Faire en sorte que sur ma classe personnage, la propriété \$vie soit égale à 50
- Dans la classe dragon, la propriété \$vie devra être redéfinie pour être égale à 100
- sur index.php , afficher la valeur de la propriété \$vie pour l'instance de la classe personnage
- sur index.php, afficher la valeur de la propriété \$vie pour l'instance de la classe dragon

- Créer une nouvelle classe ayant pour nom "princesse" dans un nouveau fichier appelé princesse.php que vous placerez dans le dossier classes
- Cette classe doit hériter de la classe personnage
- N'oubliez pas de la require à la suite dans le fichier index.php
- Vous ajouterez une propriété à cette classe, \$saved qui par défaut vaudra 0 ou false, celle ci aura une portée "private"
- Vous redéfinirez également les propriétés x et y de cette classe lorsqu'elle est instanciée, par défaut \$x = 450, \$y = 450

Théorie de l'exercice

L'intérêt majeur de la programmation orientée objet est le fait de pouvoir étendre les classes, même si le code d'origine n'a pas été réalisé par nous.

Cela permet d'ajouter des fonctionnalités à un code existant sans avoir à modifier le code d'origine, c'est très utile si vous

travaillez à plusieurs sur un même projet.

En PHP, lorsqu'on étend une classe, la classe "fille" va hériter les propriétés et les méthodes de la classe "mere"

Pour étendre une classe, on utilise le mot clef "extends", la syntaxe est la suivante

```
class Mere {...}
```

```
class Fille extends Mere { ... }
```

Pour cet exercice, j'ai repris l'exercice précédent PHP-OOP, vous remarquerez dans le dossier classe, une nouvelle classe

appelée "dragon.php", cette classe va hériter des propriétés de la classe personnage.

Souvenez vous, les propriétés définies en "private" ne sont pas héritées, j'ai donc modifié la propriété id de ma classe personnage en protected car je souhaite que les classes enfants héritent de cette propriété.

Dans ma classe dragon, je vais redéfinir la méthode setNom de ma classe personnage, je souhaite que la propriété "nom" de mon instance dragon soient sous cette forme : "Dragon <id>"

Comme ma méthode getNom est héritée, je n'ai pas besoin de la redéfinir pour pouvoir l'utiliser dans ma classe enfant "dragon"

Dans cet exercice, il vous est demandé de créer une classe qui hérite de la classe personnage, mais il aurait été possible de créer une nouvelle classe qui hérite de la classe dragon, dans ce cas celle ci aurait hérité des méthodes de la classe personnage mais aussi de la classe dragon.

Il n'y a pas de limite à l'héritage, on peut étendre une classe fille d'une classe fille etc... (mais ça devient vite trèscompliqué à gerer ;)