

Exercice

Constantes

Consignes de l'exercice

- Compléter le code de la méthode `getNumInstances` de la classe `personnage` pour que cette méthode retourne la propriété

statique `$numInstances`

- Afficher dans la page `index.php` le nombre d'instances de la classe `personnage` en utilisant la méthode appropriée de la classe `personnage`

- La classe `dragon` n'utilise pas le constructeur de la classe `personnage`, vous allez implémenter une variable statique privée

- Vous allez incrémenter cette propriété dans le constructeur comme cela a été fait pour la classe `personnage`

- Puis vous allez créer une méthode statique qui retourne le nombre de dragons créés

- Vous allez afficher le nombre de dragons créés dans la page `index.php` en utilisant la méthode que vous venez de créer

Souvenez-vous que pour appeler une méthode statique, vous devez faire référence à la classe correspondante.

Théorie de l'exercice

Les constantes sont des propriétés spéciales dont la valeur ne change pas, contrairement aux variables.

En PHP, nous pouvons définir des constantes globales (avec `define()`) et nous pouvons également définir des constantes de classe avec "const"

Exemple :

```
const DEGATS_MIN = 1;
```

Pour simplifier la lecture les constantes sont écrites en majuscules et sans le signe \$, cela permet de les différencier

plus facilement des autres propriétés.

J'ai ajouté une constante à ma classe personnage.

Je souhaite que toutes les instances de la classe personnage et toutes les classes qui héritent de cette classe puisse accéder

à cette valeur commune qui ne sera jamais modifiée.

Pour y accéder à l'intérieur de la classe personnage , j'utiliserais `self::DEGATS_MIN` , pour y accéder dans les classes

"filles" , j'utiliserais `parent::DEGATS_MIN`

Je peux également y accéder en dehors de toute classe en utilisant `personnage::DEGATS_MIN`

J'ai utilisé la constante `DEGATS_MIN` dans la méthode `attack` de ma classe `personnage`

- Le mot clef `static`

En utilisant le mot clef `static` pour déclarer une propriété, j'indique à PHP que cette valeur doit être mémorisée et propre

à la classe, je vais pouvoir y accéder comme une constante mais sa valeur va pouvoir être modifiée.

Cas concret : Je souhaite connaître le nombre d'instances de personnage qui ont été créé (ceci inclu également les

classes enfants qui font appel au constructeur parent)

J'ai donc ajouté "protected static \$numInstances" à ma classe personnage.

Dans le constructeur de ma classe personnage, j'incrémente la propriété \$numInstances à chaque fois que le constructeur est appelé

J'ai également créé une nouvelle méthode à la classe personnage afin de retourner la valeur de \$numInstances

J'ai utilisé le mot clef public static function pour déclarer cette méthode car je souhaite y accéder en faisant référence à la classe en elle même.