

Exercice

POO / OPP

Consignes de l'exercice

- Compléter le code de la classe personnage pour ajouter les méthodes suivantes :

walkLeft() (déplace le personnage vers la gauche)

walkTop() (déplace le personnage vers le haut)

walkBottom() (déplace le personnage vers le bas)

- Dans ma classe, la propriété nom est accessible directement , complétez le code de ma classe pour utiliser un getter/setter

afin de définir la propriété nom

- Définir la propriété nom sur private, le code présent sur index.php doit produire une erreur

- Modifier le code présent sur index.php qui modifie ou accède à la propriété nom pour que celui ci utilise ma fonction

setter ou getter.

Théorie de l'exercice

La programmation orienté objet en PHP vous permet de rendre votre code plus facile à modifier et de moins vous répéter dans le code.

En php, pour définir des objets, on utilise le mot clef class.

La bonne pratique consiste à créer un nouveau fichier php pour chaque classe, il est également recommandé de créer un dossier classes et d'y mettre tout les fichiers de classe php.

Une classe possède des propriétés et des méthodes.

Pour déclarer les propriétés , on utilisera les mots clefs suivant :

public \$var

private \$var

protected \$var

- Lorsqu'une proriété est délarée avec public :

La propriété est accessible partout, même dans les classes héritées , aussi bien en lecture qu'en écriture

- Lorsqu'une propriété est déclarée avec private :

La propriété est accessible seulement à l'intérieur de la classe.

- Lorsqu'une propriété est déclarée avec `protected` :

La propriété est accessible seulement à l'intérieur de la classe mais les classes qui héritent de la classe mère peuvent y accéder.

Pour déclarer les méthodes, on utilisera les mêmes mots clefs, le comportant est similaire, une méthode `private` ne sera pas utilisable en dehors de la classe, une méthode `public` le sera.

- Créer une instance de classe :

Pour créer une instance de classe, on utilise le mot clef `new` suivi du nom de la classe puis des ()

```
new nom_de_la_classe();
```

Une instance de classe est en réalité une copie unique de la classe instanciée, vous pouvez modifier directement les propriétés de l'instance sans modifier les propriétés de la classe principale.

- Constructeur de classe :

Dans la classe personnage.php, j'ai déclaré un constructeur de classe.

On déclare un constructeur de classe en utilisant cette syntaxe :

```
public function __construct()  
{  
  
}
```

Le constructeur doit toujours être défini sur public, il doit toujours être accessible en dehors de la classe.

Lorsqu'un constructeur est déclaré, cela permet de définir des propriétés ayant des valeurs par défaut qui seront

transmis automatiquement aux instances de la classe.

- Accéder aux propriétés et aux méthodes d'une instance de classe :

Pour accéder aux propriétés et aux méthodes de l'instance, on utilisera la syntaxe suivante :

```
$var->propriete
```

La syntaxe à retenir ici est l'utilisation de "->", celle-ci est obligatoire pour les instances

- Mutator ou Setter :

Une pratique courante en PHP est de définir une fonction mutator (aussi appelé setter) afin de pouvoir modifier une propriété déclarée

avec protected ou private.

Pour ce faire, il suffit de déclarer une fonction public puis définir la propriété à l'aide d'un parametre de fonction

- Accessor ou Getter :

De la même façon, pour lire un attribut privé ou protégé, il est courant de définir un getter qui va retourner la valeur

L'interet principal des getter/setter réside dans le fait que vous définissez une méthode standardisée pour définir ou lire

une propriété, si plus tard vous devez modifier la façon dont la propriété est lue ou la façon dont elle est définie,

vous n'avez que deux fonctions à modifier quelque soit le nombre de fois où vous définissez ou accédez à cette propriété

dans votre code.