

Révision Partie "Initiation à la programmation orientée objet"

Les définitions demandées sont à écrire avec vos mots, je n'ai pas besoin d'une définition toute faite trouvée sur internet.

Récapitulatif des questions :

Répondez aux questions dans un fichier .php

1. Quels termes sont utilisés dans une classe à la place de variables et fonctions? 1pt

méthodes et attributs

méthodes : les fonctions contenues dans notre classe (this->getName(), this->doSomething())

attributs : les variables attachées à notre classe (this->name, this->color etc)

2. Expliquez la différence entre "private", "protected" et "public". 2pt

private : méthodes & attributs accessible uniquement dans notre classe

protected : méthodes & attributs accessible uniquement dans notre classe et dans les classes héritées

public : accessible partout

3. En quoi consiste "l'encapsulation"? 3pt

C'est le fait de cacher (encapsuler) des attributs et méthodes pour s'assurer qu'elles ne peuvent être utilisées autrement que voulu. C'est mettre le contenu de la classe dans une boîte noire.

4. Quelle est la différence entre une classe et une interface? Ecrivez le code nécessaire pour créer la classe MyClass et l'interface MyInterface toutes les deux avec la méthode my_method() 3pt

L'interface n'aura que les signatures des méthodes la ou la classe implémente les dites méthodes.

5. A quoi sert une classe abstraite? 2pt

Une classe abstraite ne peut pas être instanciée directement. Ce qui nous permet de définir un comportement "général" tout en s'assurant que personne ne puisse l'instancier

Ex: une classe animal qui est abstraite avec des méthodes du genre "manger", "boire" et des classes filles "Lion", "Chien", "Loup". On ne pourra pas faire de new Animal mais des new Loup, new Chien etc

6. Est-ce qu'une classe peut hériter de plusieurs classes? 1pt

Non, du moins pas directement, 1 classe a maximum 1 classe mère

7. Est-ce qu'une classe peut implémenter plusieurs interfaces? 1pt

Oui, c'est d'ailleurs ce qui nous permet de faire de la composition

8. Donner un exemple d'utilisation d'une interface. 2pt

Une interface CRUD qui nous oblige à implémenter create read update et delete

Toutes les classes qui vont implémenter notre interface seront obligées d'avoir ses 4 méthodes.

9. Écrivez de 4 manières différentes la méthode "my_method()". 3pt

a.

public function my_method() {}

b.

private function my_method() {}

c.

protected function my_method() {}

d.

abstract private function my_method() {}

10. En quoi consiste la surcharge d'une méthode? 3pt

Nous permet de définir plusieurs méthodes du même nom mais avec un comportement différent

13. Peut-on empêcher la redéfinition d'une méthode? Si oui, montrez un exemple en code. 2pt

Oui avec le mot clef final

14. Pourquoi/En quoi est-ce que l'héritage peut poser problème? 3pt

**L'héritage représenté par le "IS A" (est un) (un chien EST UN animal) a des limitations ;
Une seule classe mère : donc si notre chien est un animal et aussi un être vivant on doit
soit faire : un animal EST UN être vivant et puis un chien EST UN animal qui EST UN être
vivant.**

**On est bloqué linéairement et toute modification à animal ou être vivant aura des
répercussions directes.**

**La composition a pour but d'utiliser un autre lien, le "HAS A" (a un). On va profiter du fait
qu'une classe puisse contenir d'autres classes :**

Un appartement à vendre A UN agent immobilier et A UN type de bâtiment et A UN [...]

17. Quel méthode de la classe est appelée lorsqu'un objet est instancié (new XYZ())? 1pt

Le constructeur

18. A quoi sert un getter et un setter? 1pt

Le getter permet de lire un attribut protégé de la classe

Le setter permet de modifier un attribut protégé de la classe

19. Écrivez un getter et un setter ci-dessous pour l'attribut "name". 1pt

```
public function getName() {  
    return $this->name;  
}  
public function setName($name) {  
    $this->name = $name;  
}
```

20. Est-ce que l'on peut instancier des objets d'autres classes à l'intérieur d'une classe? 2pt

Oui bien sûr

21. Quelle(s) classe(s) et/ou interface(s) créeriez-vous si l'on vous demande de réaliser le code nécessaire pour une app de gestion de biens immobiliers? 8pt

Réponse libre : ce qui est important ici c'est d'arriver à déduire les classes, l'héritage et composition qui découlerait de cet énoncé