

# POO

Les plus grandes montagnes sont faites de petites pierres...

NOM Prénom \*

Correction

Test POO, pas trop méchant.

1 - Qu'est-ce que la programmation orientée objet \*

1 point

- ☐ Un langage de programmation
- ☒ Un paradigme de programmation
- ☐ Une bibliothèque de fonctions

2 - Quel est l'avantage principal de la POO \*

1 point

- ☐ Une meilleure performance
- ☐ Une syntaxe plus simple
- ☒ Une meilleure organisation et réutilisabilité du code

3 - Qu'est-ce qu'une classe en POO ? \*

1 point

- ☐ Un objet créé à partir d'une instance
- ☐ Une fonction utilisée pour effectuer des calculs
- ☒ Un modèle pour créer des objets

4 - Qu'est-ce qu'un objet en POO ? \*

1 point

- ☒ Une instance d'une classe
- ☐ Une variable contenant une chaîne de caractères
- ☐ Une fonction utilisée pour afficher des données

5 - Qu'est-ce que l'encapsulation en POO ? \*

1 point

- ☒ Le regroupement de données et de méthodes dans une classe
- ☐ L'utilisation de fonctions préexistantes pour accomplir une tâche
- ☐ L'organisation de classes en hiérarchies pour l'héritage

6 - Qu'est-ce que l'héritage en POO \*

1 point

- ☐ La création d'une copie exacte d'une classe existante
- ☒ La capacité d'une classe à hériter des propriétés et méthodes d'une autre classe
- ☐ L'utilisation d'une interface pour définir les méthodes d'une classe

7 - Qu'est-ce qu'une méthode en POO ? \*

1 point

- ☐ Une variable contenant une valeur
- ☒ Une fonction associée à une classe
- ☐ Une classe spéciale utilisée pour stocker des données

8 - Qu'est-ce qu'une propriété en POO \*

1 point

- ☐ Une fonction utilisée pour afficher des données
- ☐ Un autre nom pour une méthode
- ☒ Une variable associée à une classe

9 - Qu'est-ce que le polymorphisme en POO ? \*

1 point

- ☒ La création d'une méthode avec le même nom mais des paramètres différents
- ☐ La capacité d'une classe à accéder à ses propres méthodes et propriétés
- ☐ L'utilisation de méthodes magiques pour modifier le comportement d'une classe

10 - Qu'est-ce qu'une interface en POO ? \*

1 point

- ☐ Une représentation graphique d'une classe
- ☐ Un modèle pour créer des objets
- ☒ Un contrat qui définit les méthodes qu'une classe doit implémenter

11 - Qu'est-ce qui permet de déterminer la visibilité d'une propriété ou d'une méthode en PHP ? \*

1 point

- ☐ public
- ☐ private
- ☐ protected
- ☒ Toutes les options sont correctes. La visibilité détermine l'accès aux propriétés et aux méthodes dans une classe.

12 - Qu'est-ce que l'autoloading en PHP ? \*

1 point

- ☒ Le chargement automatique des classes lorsqu'elles sont utilisées pour la première fois.
- ☐ L'importation automatique de bibliothèques externes dans un projet PHP.
- ☐ L'exécution automatique des méthodes lorsqu'un objet est instancié.

13 - Qu'est-ce que la méthode magique "\_\_construct()" en PHP ? \*

1 point

- ☐ Une méthode utilisée pour détruire un objet après son utilisation.
- ☐ Une méthode utilisée pour créer une copie d'un objet existant.
- ☒ Une méthode spéciale appelée automatiquement lors de l'instanciation d'un objet.

14 - Lequel des éléments suivants peut être utilisé pour instancier un objet en PHP en \* 1 point supposant que le nom de la classe soit « MaClass » ?

- ☐ \$obj = new \$maClass;
- ☒ \$obj = new MaClass();
- ☐ \$obj = new MaClass;
- ☐ obj = new MaClass();

15 - Comment invoquer une méthode? \*

1 point

- ☒ \$object->maMethod();
- ☐ object->maMethod();
- ☐ object::maMethod();
- ☐ \$object::maMethod();

16 - Comment invoquer une méthode static? \*

1 point

- ☐ object::maMethod();
- ☐ MaClass->maMethod();
- ☒ MaClass::maMethod();
- ☐ static::maMethod();

17 - Que sont les namespaces ? \*

1 point

- ☐ Un namespace est un mécanisme permettant de réduire la visibilité d'une variable ou d'une fonction à l'intérieur d'une classe.
- ☐ Un namespace est un concept permettant de gérer l'allocation et la désallocation de la mémoire.
- ☒ Un namespace est une manière d'organiser les éléments dans un programme en les regroupant dans des espaces distincts afin d'éviter les conflits de noms.
- ☐ Un namespace est un moyen d'accéder aux fonctions et méthodes d'une bibliothèque externe sans l'importer dans le code.

18 - Comment reconnaît-on une méthode magique en programmation ? \*

1 point

- ☐ Une méthode magique est une méthode qui peut être appelée uniquement par d'autres méthodes de la même classe.
- ☐ Une méthode magique commence toujours par le préfixe "magic\_" suivi du nom de la méthode.
- ☐ Une méthode magique est identifiée par le symbole @ devant son nom, par exemple @method.
- ☒ Une méthode magique a un nom spécial encadré par deux double soulignements, comme `__construct()`.

19 - Comment empêche-t-on l'héritage d'une classe en programmation ? \*

1 point

- ☒ En définissant la classe avec le mot-clé "final", par exemple : "final class MaClasse {}".
- ☐ En définissant la classe avec le mot-clé "sealed", par exemple : "sealed class MaClasse {}".
- ☐ En déclarant explicitement les méthodes et attributs de la classe comme privés.
- ☐ En ajoutant l'annotation "@noextend" avant la définition de la classe.

20 - Combien d'interfaces une classe peut-elle implémenter en PHP ? \*

1 point

- ☐ Une classe peut implémenter au maximum 3 interfaces.
- ☒ Il n'y a aucune limite au nombre d'interfaces qu'une classe peut implémenter.
- ☐ Une classe peut implémenter une seule interface à la fois.
- ☐ Une classe peut implémenter jusqu'à 5 interfaces.

Merci de me faire un retour sur la semaine, sur ce que vous avez pensé de ce module.  
Aucune vexation, au contraire les critiques aussi servent!  
(Ce formulaire est pour moi pas pour Doranco)

---

Ce contenu n'est ni rédigé, ni cautionné par Google.

Google Forms