

Programmation Orientée Objet (POO):

Voici un cahier de charge :

Une entreprise de la place souhaite en place une application web en utilisant le langage PHP (langage conçu pour le web) avec le **paradigme de la programmation orientée objet**. Vue l'envergure de ladite application, l'entreprise souhaite partager les tâches entre les groupes de développeurs et votre groupe (**groupe 1**) fait parti. Pour votre mission regarde la partie mentionnant le numéro de votre groupe (**groupe 1**) et le cadre coloré en vert comme couleur de fond.

GROUPE 7 :

Recherche :

1. Faire des recherches sur Les constantes de classes
2. Savoir maîtriser Late Static Binding
3. Comprendre comment les méthodes magiques fonctionnent : `__get()`, `__set()`, `__call()`, `__callStatic()`, `__invoke()`, `__toString()`

TAF (Travail A Faire) :

En reprenant le code du **GROUPE 2** mais toutes les propriétés doivent être en privée puis suivre ces étapes suivantes :

1. Définir la méthode `__construct()` en utilisant la technique de la promotion du constructeur qui vont initialiser les propriétés de la classe **BankAccount**
2. Dans cette même classe, définissez les méthodes magiques `__get()`, `__set()` et `__call()` pour gérer l'accès aux propriétés et aux méthodes de la classe. La méthode `__get()` doit être appelée automatiquement lorsque l'on tente d'accéder à une propriété qui **n'existe pas** ou qui est **protégée (protected)** ou **privée(private)**. Dans notre cas, il faut qu'elle renvoie simplement la valeur de la propriété si elle existe. Quant à la méthode `__set()` elle doit être appelée automatiquement lorsque l'on tente de modifier la valeur d'une propriété qui

n'existe pas ou qui est **protégée (protected)** ou **privée (private)**. Dans notre exemple, elle doit modifier simplement la valeur de la propriété si elle existe. Enfin, la méthode **__call()** doit être appelée automatiquement lorsque l'on tente d'appeler une méthode qui **n'existe pas** ou qui est **protégée (protected)** ou **privée (private)**. Dans notre exemple, elle va permettre de définir un comportement personnalisé pour les méthodes **deposit()** et **withdraw()**. La méthode **deposit()** ajoute le montant spécifié au solde du compte, tandis que la méthode **withdraw()** retire le montant

Fin du GROUPE 7

GROUPE 8 :

Recherche :

1. Sérialisation des Objets avec la méthode **serialize()**
2. Désérialisation des objets avec la méthode **unserialize()**
3. clonage des objets
4. Comparaison des objets
5. Les classes anonymes

TAF (Travail A Faire) :

Dans le code du **groupe 7**, on s'est contenté de définir notre classe **BankAccount** sans instancier les objets de cette classe et c'est ce que nous allons faire dans ce groupe mais en travaillant plus sur les objets. Voici ce que vous devez faire :

1. Instancier d'abord la classe **BankAccount** et nommé cet objet **\$compte**
2. Sérialiser cet objet en utilisant la méthode **serialize()** qui permet de transformer l'objet en une chaîne de caractères et stocké ça dans la variable **\$unserialized**
NB : Il est important de noter que pour utiliser la méthode **serialize()**, les propriétés de l'objet doivent être accessibles (c'est-à-dire **publics** ou avec des méthodes **d'accès**).
3. Afficher la chaîne de caractères sérialisée
4. Désérialiser ensuite cette chaîne de caractères en un objet **\$compte** avec la méthode **unserialize()**
5. Ré-afficher l'objet **\$compte** désérialisé
6. Instancier encore la classe **BankAccount** et nommé cet objet **\$compte1**
7. Cloner cet objet **\$compte1** et nommé le nouveau objet **\$compte2**
8. Modifier le solde du **\$compte2**
9. Afficher les soldes des deux objets **\$compte1** et **\$compte2**
10. Que remarquez vous ?
11. Utiliser les deux (02) façons (**==** et méthode **equals()**) pour comparer les objets **\$compte1** et **\$compte2**
12. Comment insérer alors la classe **anonyme** pour modifier le comportement de votre classe **BankAccount** ?

Fin du GROUPE 8

GROUPE 9 :

Recherche :

1. Les namespaces
2. Le mot clé use dans namespaces
3. le mot clé alias

TAF (Travail A Faire) :

1. Reprenez le code de n'importe quel groupe et ajouter les **namespaces**
2. Après l'ajout des **namespaces** ,utiliser le mot clé **use** + **namespace** pour mettre en valeur le mot clé use d'une part et d'autre part appliquer aussi le mot clé **alias** puis dites nous dans quel cas qu'on utilise les **alias**.

Fin du GROUPE 9