

Exercices Pratiques de Python : Programmation Orientée Objet

Ali ZAINOUL

Exercices de Programmation Orientée Objet (POO)

1. Classe Person :

- Créez une classe **Person** avec les attributs suivants : **name** et **age**.
- Ajoutez une méthode **display()** qui affiche le nom et l'âge de la personne.
- Instanciez un objet de la classe **Person** et appelez la méthode **display()**.

2. Classe Rectangle :

- Créez une classe **Rectangle** avec les attributs suivants : **length** et **width**.
- Ajoutez une méthode **area()** qui retourne l'aire du rectangle.
- Ajoutez une méthode **perimeter()** qui retourne le périmètre du rectangle.
- Instanciez un objet de la classe **Rectangle** et affichez son aire et son périmètre.

3. Classe BankAccount :

- Créez une classe **BankAccount** avec les attributs suivants : **account_number** et **balance**.
- Ajoutez une méthode **deposit(amount)** qui ajoute un montant au solde.
- Ajoutez une méthode **withdraw(amount)** qui soustrait un montant du solde.

- Ajoutez une méthode `display()` qui affiche le numéro de compte et le solde.
- Instanciez un objet de la classe `BankAccount` et effectuez des dépôts, des retraits et affichez le solde.

4. Classe Car :

- Créez une classe `Car` avec les attributs suivants : `make`, `model` et `year`.
- Ajoutez une méthode `display()` qui affiche la marque, le modèle et l'année de la voiture.
- Ajoutez une méthode `start()` qui affiche un message indiquant que la voiture démarre.
- Instanciez un objet de la classe `Car` et appelez les méthodes `display()` et `start()`.

5. Getters et Setters :

- Créez une classe `Circle` avec un attribut privé `_radius`.
- Ajoutez des méthodes getter et setter pour l'attribut `radius`.
- Ajoutez une méthode `area()` qui retourne l'aire du cercle.
- Instanciez un objet de la classe `Circle`, modifiez le rayon en utilisant le setter et affichez l'aire.

6. Classe Employee :

- Créez une classe `Employee` avec les attributs privés `_name` et `_salary`.
- Ajoutez des méthodes getter et setter pour les attributs `name` et `salary`.
- Ajoutez une méthode `display()` qui affiche le nom et le salaire de l'employé.
- Instanciez un objet de la classe `Employee`, modifiez le salaire en utilisant le setter et affichez les informations de l'employé.

7. Classe Product :

- Créez une classe `Product` avec les attributs privés `_name` et `_price`.
- Ajoutez des méthodes getter et setter pour les attributs `name` et `price`.

- Ajoutez une méthode `apply_discount(discount)` qui applique une réduction au prix du produit.
- Instanciez un objet de la classe `Product`, appliquez une réduction et affichez le prix mis à jour.