

# Série d'Exercices de Programmation Orientée Objet Avancée

## Exercice 1 : Héritage Multiple

Définir deux classes de base, `Vehicle` et `Watercraft`.

- La classe `Vehicle` doit avoir un attribut `wheels` (nombre de roues).
- La classe `Watercraft` doit avoir un attribut `propellers` (nombre d'hélices).
- Créer une classe `AmphibiousVehicle` qui hérite des deux classes et initialise les deux attributs.

### Instructions supplémentaires :

- Implémentez les constructeurs pour initialiser les attributs dans chaque classe.
- Ajoutez une méthode `display` dans `AmphibiousVehicle` pour afficher le nombre de roues et d'hélices.

## Exercice 2 : Mixins

Créer un mixin `SerializableMixin` qui ajoute une méthode de sérialisation à n'importe quelle classe qui l'inclut.

- La méthode de sérialisation doit convertir les attributs de l'objet en un dictionnaire.
- Créer une classe `Book` avec des attributs `title` et `author`, et faites en sorte qu'elle hérite de `SerializableMixin`.
- Ajouter une méthode `to_json` pour convertir le dictionnaire en chaîne JSON.

### Exercice 3 : Refactorisation avec Composition

Refactoriser une classe `Bird` qui utilise l'héritage pour obtenir des comportements de vol, en utilisant la composition à la place.

- Créer une classe `FlyBehavior` avec une méthode `fly`.
- Créer une classe `Bird` qui utilise une instance de `FlyBehavior` au lieu d'hériter d'une classe de vol.
- Ajouter une méthode `set_fly_behavior` pour changer le comportement de vol dynamiquement.

### Exercice 4 : Composition pour des Personnages de Jeu

Concevoir un système utilisant la composition pour construire des personnages de jeu avec des capacités modifiables.

- Créer des classes `FightingAbility` et `FlyingAbility` avec des méthodes `fight` et `fly`.
- Créer une classe `GameCharacter` qui peut inclure différents comportements en utilisant la composition.
- Ajouter des méthodes pour modifier les capacités des personnages en cours de jeu.

### Exercice 5 : Classe `APIConsumer`

Écrire une classe `APIConsumer` utilisant la bibliothèque `requests` pour récupérer des données météorologiques d'une API publique.

- Utiliser une API comme `OpenWeatherMap`.
- Implémenter une méthode `fetch_weather(city)` pour obtenir les données météorologiques d'une ville donnée.
- Traiter les réponses de l'API pour afficher des informations comme la température, l'humidité et les conditions météorologiques.

### Exercice 6 : Classe `ImageEditor`

Créer une classe `ImageEditor` utilisant la bibliothèque `Pillow` pour effectuer des opérations de base sur les images.

- Ajouter des méthodes pour la rotation (`rotate(angle)`), le redimensionnement (`resize(width, height)`) et la conversion en niveaux de gris (`convert_to_grayscale`).
- Charger une image depuis un fichier et appliquer ces opérations.

## Exercice 7 : Classe User avec SQLAlchemy

Définir une classe `User` et utiliser `SQLAlchemy` pour mapper cette classe à une table de base de données.

- La classe doit avoir des attributs `id`, `name`, et `email`.
- Configurer la base de données et créer la table correspondante.
- Ajouter des méthodes pour créer et récupérer des utilisateurs.

## Exercice 8 : Méthodes CRUD dans User

Écrire des méthodes dans la classe `User` pour ajouter, modifier, et supprimer des utilisateurs de la base de données.

- Ajouter une méthode `add_user(name, email)` pour ajouter un nouvel utilisateur.
- Ajouter une méthode `update_user(id, name, email)` pour modifier un utilisateur existant.
- Ajouter une méthode `delete_user(id)` pour supprimer un utilisateur.

## Exercice 9 : Nettoyage de Données

Charger un jeu de données CSV dans un `DataFrame` Pandas et effectuer un nettoyage de données.

- Supprimer les valeurs nulles.
- Normaliser les noms de colonnes (par exemple, tout en minuscules, remplacement des espaces par des underscores).
- Afficher un aperçu des données avant et après le nettoyage.

## Exercice 10 : Statistiques Descriptives

Utiliser Pandas pour calculer des statistiques descriptives sur un ensemble de données spécifique.

- Calculer la moyenne, la médiane et le mode de colonnes numériques.
- Afficher les résultats de ces calculs.

## Exercice 11 : Histogramme des Âges

Créer un histogramme des âges des utilisateurs à partir d'un DataFrame Pandas.

- Charger un DataFrame contenant des données d'utilisateurs.
- Utiliser `matplotlib` pour générer l'histogramme des âges.
- Ajouter des titres et des labels aux axes.

## Exercice 12 : Diagramme à Dispersion

Générer un diagramme à dispersion (scatter plot) pour analyser la corrélation entre deux variables dans un DataFrame.

- Choisir deux colonnes numériques dans un DataFrame.
- Utiliser `matplotlib` pour créer le scatter plot.
- Ajouter une ligne de tendance si possible.

## Exercice 13 : Scraping de Données

Développer un script pour scraper les données des nouvelles du jour d'un site web et les enregistrer dans un fichier CSV.

- Utiliser des bibliothèques comme `BeautifulSoup` ou `Scrapy`.
- Extraire des informations pertinentes (titre, date, lien) et les enregistrer dans un fichier CSV.

## Exercice 14 : Application de Gestion de Contacts

Construire une petite application de gestion de contacts en utilisant `SQLAlchemy`, où les contacts peuvent être ajoutés, supprimés, et mis à jour dans une base de données.

- Créer une interface simple pour interagir avec l'utilisateur.
- Implémenter des fonctionnalités pour ajouter, modifier et supprimer des contacts.
- Assurer la persistance des données dans une base de données.