

Module : POO Python (M103) TP N° 2

Année de Formation 2023/2024

Filière : Développement digital

Groupe : DEV 101 - DEV 102

Niveau : 1ère année

Exercice 1 :

Créez une classe de base **Animal** avec les attributs **nom** et **espece**.

La méthode **parler** doit être définie dans la classe **Animal**, mais elle ne fait rien (utilisez `print("je parle")`).

1- Créez trois classes dérivées : **Chien**, **Chat** et **Vache**. Chacune de ces classes doit hériter de la classe **Animal** et **redéfinir** la méthode **parler** pour retourner le son spécifique de l'animal (par exemple, "**Haw!**" pour un chien, "**Meow!**" pour un chat, "**Moo!**" pour une vache).

2- Créez des instances de chaque classe d'animaux avec des noms et des espèces différents.

3- Créez une liste appelée **animaux** contenant ces instances.

4- Parcourez la liste **animaux** et appelez la méthode **parler** pour chaque animal. Affichez le **nom** de l'animal, son **espèce** et le **son** qu'il émet.

Exercice 2 :

1. Créez une classe de base **Livre** avec les attributs suivants :

- **titre**
- **auteur**
- **annee_publication**
- **identifiant** (un numéro unique pour chaque livre)

2. Créez une classe dérivée **`LivreEmpruntable`** qui hérite de la classe **`Livre`**.
Ajoutez un attribut booléen **`est_emprunte`** pour indiquer si le livre est actuellement emprunté ou non.
3. Implémentez les méthodes suivantes dans la classe **`LivreEmpruntable`** :
 - **`emprunter()`** : Cette méthode devrait marquer le livre comme emprunté.
 - **`retourner()`** : Cette méthode devrait marquer le livre comme retourné.
4. Créez une classe **`Bibliotheque`** qui contient une liste de livres empruntables.
5. Implémentez une méthode **`afficher_livres()`** dans la classe **`Bibliotheque`** qui affiche tous les livres, en indiquant si chaque livre est emprunté ou non.
6. Créez des instances de la classe **`LivreEmpruntable`** et ajoutez-les à la bibliothèque. Empruntez quelques livres et retournez-les pour tester le fonctionnement de votre code.

Exercice 3 :

1. Créez une classe de base **`Employe`** avec les attributs suivants :
 - **`nom`**
 - **`salaire`**
2. Créez deux classes dérivées : **`Chef`** et **`EmployeOrdinaire`**. **`Chef`** hérite de **`Employe`** et a un attribut supplémentaire **`service`** pour indiquer le service auquel le chef est associé.
3. Ajoutez une méthode **`afficher_details`** à la classe de base **`Employe`** qui affiche le **nom** et le **salaire** de l'employé.
4. Ajoutez une méthode **`afficher_details`** à la classe **`Chef`** qui affiche les détails de l'employé ainsi que le service.

5. Créez une classe **`Entreprise`** qui contient une liste d'employés (à la fois des chefs et des employés ordinaires).
6. Ajoutez une méthode **`afficher_employes`** à la classe **`Entreprise`** qui affiche les détails de tous les employés, en appelant la méthode **`afficher_details`** appropriée en fonction du type d'employé.
7. Créez des instances d'employés, à la fois des chefs et des employés ordinaires, et ajoutez-les à l'entreprise.
8. Appelez la méthode **`afficher_employes`** pour afficher les détails de tous les employés de l'entreprise.

Exercice 4 :

1. Créez une classe de base **`Fichier`** avec les attributs suivants :
 - **`nom`** (le nom du fichier)
 - **`taille`** (la taille du fichier en octets)
 - **`date_creation`** (la date de création du fichier)
2. Créez une classe dérivée **`Dossier`** qui hérite de la classe **`Fichier`**. Les dossiers sont des fichiers spéciaux qui peuvent contenir d'autres fichiers (fichiers ou dossiers).
3. Ajoutez une méthode **`afficher_info`** à la classe **`Fichier`** qui affiche le nom, la taille et la date de création du fichier.
4. Ajoutez une méthode **`afficher_info`** à la classe **`Dossier`** qui affiche les informations du dossier, y compris son nom, sa taille (qui est la somme des tailles de tous les fichiers et sous-dossiers qu'il contient) et la date de création du dossier.

5. Créez un dossier racine, puis ajoutez des fichiers et des sous-dossiers à la structure. Assurez-vous de créer une structure de dossiers et de fichiers imbriqués pour tester la hiérarchie.
6. Appelez la méthode ``afficher_info`` sur le dossier racine pour afficher les informations de tout le système de fichiers, y compris les fichiers et les dossiers imbriqués.