

TP N°7 : Les collections POO

Objectif : Manipuler les listes d'objets.

Exercice 1 :

A. Créer la classe **stagiaire** caractérisée par :

1. Les attributs : CNE, Nom, Prénom
2. Les propriétés.
3. Un constructeur d'initialisation
4. La méthode `__str__()`

B. Créez une classe **Groupe** caractérisée par :

1. Les attributs : le nom du groupe, la filière, et une liste de stagiaires.
2. Les propriétés.
3. Un constructeur d'initialisation
4. La méthode `__str__()`
5. Une méthode qui permet de rechercher un stagiaire par son CNE dans le groupe.
6. Une méthode qui vérifie si un stagiaire spécifique (CNE) existe dans le groupe.
7. Une méthode qui vérifie si un stagiaire spécifique (Nom et prénom) existe dans le groupe.
8. Une méthode qui permet d'ajouter un stagiaire au groupe (vérifiez que le CNE est unique et qu'il n'y a pas déjà 25 stagiaires dans le groupe).
9. Une méthode qui permet d'afficher la liste des stagiaires dans le groupe.
10. Une méthode qui permet de retirer un stagiaire du groupe à partir de son CNE.

C. Testez votre classe en créant un groupe, en ajoutant quelques stagiaires, en affichant la liste des stagiaires, en retirant un stagiaire, et en vérifiant que la limite de 24 stagiaires est respectée, etc.

Exercice 2 :

Soit les classes suivantes :

Client (CIN, Nom, Prénom, Adresse, Tél)

Compte (Numéro, solde, type, Client)

Banque (Nom, Siège, Liste de Comptes)

Travail à faire :

- a) Créer la classe Client.
 - Créer un constructeur d'initialisation.
 - Créer une propriété pour chaque attribut de la classe.
 - Créer la méthode `__str__()` permettant de retourner les caractéristiques d'un client.
- b) Créer la classe Compte.
 - Le numéro de compte est incrémenté automatiquement.
 - Créer un constructeur d'initialisation.
 - Créer une propriété pour chaque attribut de la classe
 - Le type d'un compte est 'C' pour compte courant ou 'E' pour compte d'épargne (sinon lever une exception)
 - Le solde est en lecture seule
 - Créer la méthode `__str__()` permettant de retourner les caractéristiques d'un compte.
- c) Créer la classe Banque.
 - Créer un constructeur d'initialisation (initialiser le nom et le siège, La liste de comptes est initialisée par une liste vide)
 - Créer une propriété pour chaque attribut.
 - Créer la méthode `__str__()` permettant de retourner les informations de la banque (Le nom, le siège, et la liste de comptes)
 - Créer la méthode `RechercherCompte(num)` permettant de retourner un compte à partir de son numéro.
 - Créer la méthode `AjouterCompte(Compte C)` permettant d'ajouter un compte donnée à la banque en cours. (Retourner un booléen)
 - Créer la méthode `SupprimerCompte(int num)` permettant de supprimer un compte à partir de son numéro. (Si son solde est égal à zéro). (Retourner un booléen)
 - Créer la méthode `ModifierTelClient(int num, string Tel)` permettant de modifier le numéro de téléphone du client d'un compte donné. (Retourner un booléen)
 - Créer la méthode `ReturnComptes()` permettant de retourner la liste de tous les comptes de la banque.
 - Créer la méthode `ReturnComptesE()` permettant de retourner la liste de tous les comptes d'épargne de la banque.
 - Créer la méthode `ReturnComptesC(min,max)` permettant de retourner la liste de

tous les comptes courants de la banque qui ont un solde entre deux valeurs données.

- Créer la méthode `ReturnComptes(cin)` permettant de retourner la liste de tous les comptes d'un client donné.
- Créer la méthode `CompteMax()` permettant de retourner les comptes ayant le solde maximal.

d) Dans le programme principal, Instancier **un Objet de la classe Banque**

(B= Banque (« BP », « CASA »))

Sous forme d'un Menu réaliser les opérations suivantes (Ex sur la banque B) en appelant les méthodes appropriées de la classe Banque :

- Ajouter un compte à la banque.
- Supprimer un compte à partir de son code (si son solde est égal à zéro).
- Modifier le numéro de téléphone du client d'un compte donné.
- Rechercher un compte à partir de son Numéro.
- Afficher toutes les informations de la banque.
- Afficher tous les comptes de la banque
- Afficher tous les comptes d'épargne de la banque
- Afficher les comptes courants qui ont un solde entre deux valeurs données.
- Rechercher le(s) compte(s) à partir de son Client (CIN).
- Afficher la somme de tous les comptes d'épargne de la banque.
- Afficher la somme de tous les comptes d'un client donné (CIN)
- Afficher les comptes ayant le solde maximal.

NB : le programme principal sera composé des appels de différentes méthodes de la classe Banque.