

## **TP7 : gestion des voyages**

L'objet de cette application est la gestion informatisée des voyages organisés par une société possédant des bus. Chaque voyage est assuré par un seul chauffeur.

1. Classe Chauffeur.
  - a. Définir le constructeur de la classe Chauffeur dont les caractéristiques sont : CIN, Nom, Prenom.
  - b. Ecrire les accesseurs des champs
  - c. Ecrire la méthode `__str__()`
2. Classe Bus
  - a. Définir le constructeur de la classe Bus dont les caractéristiques sont : Immatriculation, Marque, Type.
  - b. Ecrire les accesseurs des champs
  - c. Ecrire la méthode `__str__()`
3. Classe Voyage
  - a. Définir le constructeur de la classe Voyage dont les caractéristiques sont :
    - ✓ Numero voyage : le numéro du 1<sup>er</sup> voyage créer est 1 et à chaque création d'un nouveau voyage, ce numéro doit être automatiquement incrémenté de 1.
    - ✓ Vchauffeur : c'est le chauffeur qui a assuré le voyage courant.
    - ✓ Vbus : c'est le bus conduit par Vchauffeur au cours de ce voyage.
    - ✓ Date voyage : date où a eu lieu ce voyage.
    - ✓ Ville de départ
    - ✓ Ville d'arrivée
    - ✓ Nombre de voyageurs : nombre des personnes qui ont effectué ce voyage.
    - ✓ Prix du billet : prix payé par chaque voyageur de ce voyage (le même prix pour tous).
  - b. Ecrire les accesseurs des champs
  - c. Ecrire la méthode `__str__()` qui renverra le numéro du voyage, la date du voyage, le nom et le prénom du chauffeur, l'immatriculation et la marque du bus, ville de départ, ville d'arrivée et recette du voyage.  
la recette du voyage = Nombre de voyageurs \* prix du billet du voyage en cours.
4. Classe GestionVoyage :
  - a. Déclarer une liste de chauffeurs, une liste de bus et une liste de voyages.
  - b. Ecrire la méthode `AjouterChauffeur()` : qui permet d'ajouter un chauffeur à la liste des chauffeurs
  - c. Ecrire la méthode `AjouterBus()` : qui permet d'ajouter un bus à la liste des bus
  - d. Ecrire la méthode `AjouterVoyage()` : qui permet d'ajouter un voyage à la liste des voyages.
  - e. Ecrire la méthode `rechercherChauffeur()` qui recherche un chauffeur par son CIN. Si trouvé, elle retourne le chauffeur correspondant dans la liste des chauffeurs. Sinon, elle retourne la valeur null.
  - f. Ecrire la méthode `rechercheBus()` qui recherche un bus par son immatriculation. Si trouvé, elle retourne le bus correspondant dans la liste des bus. Sinon, elle retourne la valeur null.

- g. Ecrire la méthode rechercheVoyage() qui recherche un voyage par son numéro. Si trouvé, elle retourne l'indice correspondant dans la liste des voyages. Sinon, elle retourne la valeur -1.
- h. Ecrire la méthode AfficherVoyages() qui affiche tous les voyages
- i. Concevoir un menu pour tester les méthodes écrites.

Indications à tenir en compte :

- ✓ AjouterChauffeur() : faire le nécessaire pour que deux chauffeurs dans la liste n'aient pas le même numéro CIN
- ✓ Ajouterbus() : faire le nécessaire pour que deux bus dans la liste n'aient pas la même immatriculation
- ✓ AjouterVoyage() :
  - saisir le CIN du chauffeur qui doit correspondre à un chauffeur de la liste des chauffeurs et si trouvé, le chauffeur ainsi trouvé correspond à Vchauffeur du voyage à ajouter, sinon le programme vous demandera de saisir un autre CIN.
  - Saisir l'immatriculation du bus qui doit correspondre à un bus de la liste des bus et si trouvée, le bus ainsi trouvé correspond à Vbus du voyage à ajouter, sinon le programme vous demandera de saisir une autre immatriculation.