

Partie Théorique I : (15 pts)

Une entreprise de gestion de flotte automobile propose un service de suivi et d'optimisation des véhicules pour des clients. Chaque véhicule est enregistré avec un type spécifique (voiture, camion, moto) et a des caractéristiques propres (consommation de carburant, capacité de transport, autonomie, etc.).

L'objectif est de concevoir une application permettant à l'entreprise de gérer les véhicules de ses clients.

1. Définissez une classe abstraite `Vehicule` contenant les attributs communs (`nom_client`, `matricule`, `capacite`) et une méthode abstraite `calculer_efficiency()`. (1 pt)
2. Ajoutez une méthode d'instance `ajuster_capacite(self, valeur)` permettant d'augmenter ou de diminuer la capacité de transport d'un véhicule. Assurez-vous que la capacité ne devienne pas négative. (2 pts)
3. Implémentez la classe `Voiture` qui hérite de `Vehicule`, avec un attribut `consommation` (litres/100km) et une méthode `calculer_efficiency()` retournant l'autonomie du véhicule en fonction du carburant disponible. (2 pts)
4. Implémentez la classe `Camion` avec un attribut `charge_max` et la méthode `calculer_efficiency()` qui estime la distance pouvant être parcourue en fonction de la charge transportée. (2 pts)
5. Ajoutez une méthode magique `__str__()` dans `Vehicule` pour afficher les informations d'un véhicule sous la forme :

Client: XYZ Logistics | Matricule: 123-ABC | Capacité: 500kg (1 pt)

6. Implémentez une méthode magique `__eq__()` pour comparer deux véhicules selon le matricule. (2 pts)
7. Créez une exception `CapaciteError` qui sera levée si une capacité négative est définie. (2 pts)
8. Ajoutez des getters et setters pour l'attribut `capacite`, afin d'empêcher une valeur négative. (1 pt)
9. Modifiez `calculer_efficiency()` pour qu'elle lève une exception `NotImplementedError` si elle n'est pas redéfinie dans une sous-classe. (2 pts)

Partie Pratique II : (25 pts)

10. Créez une interface Tkinter avec : (5 pts)

- Un champ de saisie pour le matricule du véhicule.
- Un champ de saisie pour la capacité du véhicule.
- Une liste déroulante pour choisir le type de véhicule (Voiture, Camion, Moto).
- Un bouton "Ajouter" qui crée un véhicule et l'affiche dans une Listbox.

11. Ajoutez une gestion des erreurs pour empêcher d'ajouter un véhicule avec une capacité négative et afficher un message d'erreur (`messagebox.showerror`). (3 pts)

12. Ajoutez une Treeview permettant d'afficher les véhicules avec les colonnes : (5 pts)

- Matricule
- Capacité (kg ou L)
- Type de Véhicule
- Efficacité estimée

Le bouton "Ajouter" doit également mettre à jour la Treeview avec ces informations. (2 pts)

13. Ajoutez un bouton permettant de supprimer un véhicule sélectionné dans la Treeview. (5 pts)
14. Ajoutez une fonctionnalité permettant de modifier un véhicule existant (ex : changer sa capacité). (5 pts)