

## Classe Main

- **Variables:**
  - Aucune variable d'instance définie dans les commentaires du code.
- **Méthodes:**
  - `main(String[] args):` Point d'entrée de l'application.
  - `adminMenu(Scanner scanner, List<Vehicule> vehicules, List<Course> courses, GestionnaireCourse gestionnaireCourse):` Menu pour l'administrateur.
  - `clientMenu(Scanner scanner, Client currentUser, List<Vehicule> vehicules, List<Lieu> lieux, List<Course> courses, GestionnaireCourse gestionnaireCourse):` Menu pour les clients.
  - `checkAndCompleteActiveRide(Scanner scanner, Client currentUser, List<Course> courses):` Vérifie et complète une course active.
  - `terminateActiveRide(Scanner scanner, Client currentUser, List<Course> courses):` Termine une course active.
  - `modifyAccountInformation(Scanner scanner, Client currentUser):` Modifie les informations du compte client.
  - `listClientRides(String clientNumero):` Liste les courses d'un client.
  - `listAllRides():` Liste toutes les courses.
  - `getPassengersInVehicle(Vehicule vehicule, List<Course> courses):` Obtient le nombre de passagers dans un véhicule.

## Classe Utils

- **Méthodes:**
  - `getIntInputWithRetries(Scanner scanner, String prompt, int retries):` Obtient une entrée entière avec tentatives.
  - `getDoubleInputWithRetries(Scanner scanner, String prompt, int retries):` Obtient une entrée double avec tentatives.
  - `getInputWithRetries(Scanner scanner, String prompt, int retries):` Obtient une entrée de chaîne avec tentatives.
  - `ensureCsvHeaders(String fileName, String headers):` Assure la présence des en-têtes CSV.

## Classe Client

- **Variables:**
  - `nextClientID:` Identifiant de la prochaine client.
  - `clientID:` Identifiant du client.
  - `numero:` Numéro de téléphone du client.
  - `nom:` Nom du client.
  - `prenom:` Prénom du client.
  - `password:` Mot de passe du client.
- **Méthodes:**
  - `Client(String clientID, String numero, String nom, String prenom, String password):` Constructeur.
  - `generateUniqueID():` Génère un identifiant unique.
  - `getClientID():` Obtient l'identifiant du client.
  - `getNumero():` Obtient le numéro de téléphone du client.
  - `getNom():` Obtient le nom du client.

- `getPrenom()` : Obtient le prénom du client.
- `getPassword()` : Obtient le mot de passe du client.
- `setNumero(String numero)` : Définit le numéro de téléphone du client.
- `setPassword(String password)` : Définit le mot de passe du client.
- `setNom(String nom)` : Définit le nom du client.
- `setPrenom(String prenom)` : Définit le prénom du client.
- `creerCompte()` : Crée un nouveau compte client.
- `numeroAlreadyExists(String numero)` : Vérifie si le numéro de téléphone existe déjà.
- `chargerClients()` : Charge tous les clients depuis le fichier.
- `login(String numero, String password)` : Connexion pour le client.
- `updateNumero(String newNumero)` : Met à jour le numéro de téléphone du client.
- `updatePassword(String newPassword)` : Met à jour le mot de passe du client.
- `updateNom(String newNom, String newPrenom)` : Met à jour le nom du client.
- `updateClientDetails()` : Met à jour les détails du client dans le fichier.
- `supprimerClient(String clientNumero)` : Supprime un client du fichier.

## Classe Course

- **Variables:**

- `nextCourseID`: Identifiant de la prochaine course.
- `idCourse`: Identifiant de la course.
- `client`: Client associé à la course.
- `vehicule`: Véhicule associé à la course.
- `lieuDepart`: Lieu de départ.
- `lieuDarrivee`: Lieu d'arrivée.
- `timeOrdered`: Heure de commande.
- `timeArrived`: Heure d'arrivée.
- `distance`: Distance de la course.
- `amount`: Montant de la course.
- `passengers`: Nombre de passagers.
- `completed`: Statut de complétion de la course.

- **Méthodes:**

- `Course(int idCourse, Client client, Vehicule vehicule, Lieu lieuDepart, Lieu lieuDarrivee, String timeOrdered, double distance, int amount, int passengers)`: Constructeur.
- `generateUniqueID()` : Génère un identifiant unique pour la course.
- `enregistrerCourse()` : Enregistre la course dans le fichier des courses.
- `genererRecu()` : Génère un reçu pour la course.
- `chargerCourses(List<Client> clients, List<Vehicule> vehicules, List<Lieu> lieux)` : Charge toutes les courses depuis le fichier.
- `afficherDetailsCourse()` : Affiche les détails de la course.
- `afficherRecu()` : Affiche les détails du reçu.
- `terminerCourse()` : Termine la course et génère un reçu.
- `getTotalTime()` : Calcule le temps total de la course.
- `estimateArrivalTime()` : Estime l'heure d'arrivée.

- `getIdCourse()` : Obtient l'identifiant de la course.
- `getClient()` : Obtient le client associé à la course.
- `getVehicule()` : Obtient le véhicule associé à la course.
- `getLieuDepart()` : Obtient le lieu de départ.
- `getLieuDarrivee()` : Obtient le lieu d'arrivée.
- `getTimeOrdered()` : Obtient l'heure de commande.
- `getDistance()` : Obtient la distance de la course.
- `getAmount()` : Obtient le montant de la course.
- `getPassengers()` : Obtient le nombre de passagers.
- `isCompleted()` : Vérifie si la course est complétée.

## Classe **Vehicule**

- **Variables:**

- `nextVehiculeID`: Identifiant du prochain véhicule.
- `carID`: Identifiant du véhicule.
- `numeroMatricule`: Numéro de plaque du véhicule.
- `type`: Type de véhicule.

- **Méthodes:**

- `Vehicule(String carID, String numeroMatricule, String type)`: Constructeur.
- `generateUniqueID()` : Génère un identifiant unique pour le véhicule.
- `getCarID()` : Obtient l'identifiant du véhicule.
- `getNumeroMatricule()` : Obtient le numéro de plaque du véhicule.
- `getType()` : Obtient le type de véhicule.
- `setNumeroMatricule(String numeroMatricule)` : Définit le numéro de plaque du véhicule.
- `setType(String type)` : Définit le type de véhicule.
- `chargerVehicules()` : Charge tous les véhicules depuis le fichier.
- `saveVehicules(List<Vehicule> vehicules)` : Sauvegarde tous les véhicules dans le fichier.

## Classe **Lieu**

- **Variables:**

- `nom`: Nom du lieu.
- `longitude`: Longitude du lieu.
- `latitude`: Latitude du lieu.

- **Méthodes:**

- `Lieu(String nom, double longitude, double latitude)`: Constructeur.
- `getNom()` : Obtient le nom du lieu.
- `getLongitude()` : Obtient la longitude du lieu.
- `getLatitude()` : Obtient la latitude du lieu.
- `setNom(String nom)` : Définit le nom du lieu.
- `setLongitude(double longitude)` : Définit la longitude du lieu.
- `setLatitude(double latitude)` : Définit la latitude du lieu.
- `calculateDistance(Lieu start, Lieu end)` : Calcule la distance entre deux lieux en utilisant la formule de Haversine.

## **Classe GestionnaireCourse**

- **Variables:**
  - `vehicules`: Liste des véhicules.
- **Méthodes:**
  - `GestionnaireCourse(List<Vehicule> vehicules)`: Constructeur.
  - `enregistrerVehicule(Vehicule vehicule)`: Enregistre un nouveau véhicule.
  - `retirerVehicule(String numeroMatricule)`: Retire un véhicule par son numéro de plaque.
  - `listerVehicules()`: Liste tous les véhicules.