Projet de Fin de Cours
Système de Gestion d'Entreprise de Location de Véhicules

Objectif général

Mettre en pratique tous les concepts de la Technologie Orientée Objet à travers la conception et l'implémentation d'une application logicielle complète, depuis la modélisation simple jusqu'à la mise en œuvre fonctionnelle, tout en utilisant un langage orienté objet (Python votre choix de cours).

Contexte du projet

L'entreprise ACA souhaite développer un logiciel de gestion de location de véhicules pour ses agences.

Ce logiciel doit permettre :

- d'enregistrer les véhicules disponibles à la location,
- de gérer les clients et leurs contrats de location,
- de suivre les paiements,
- et de produire de petits rapports (par exemple, la liste des véhicules loués, les clients en retard, etc.).

L'objectif est de produire un système bien structuré, modulaire, maintenable et réutilisable.

Découpage du projet

Le projet se déroule en 6 étapes correspondant aux 6 chapitres du cours. Chaque étape est notée individuellement mais contribue au projet global.

➢ Étape 1 – Concepts Fondamentaux (Classes, Objets, Encapsulation, Héritage, Polymorphisme, Abstraction)

Objectif : Créer la base du modèle objet du système.

Travail demandé :

1. Créez les classes principales suivantes :
    - Vehicule (classe de base)
    - Voiture et Moto (héritant de Vehicule)
    - Client
    - ContratLocation

2. **Encapsulation :**
    - Tous les attributs doivent être privés.
    - Fournissez des **getters** et **setters**.
3. **Héritage et Polymorphisme :**
    - Voiture et Moto redéfinissent une méthode **afficherDetails()**.
    - Ajoutez une méthode abstraite **calculerTarifLocation()** dans Véhicule, et implémentez-la différemment selon le type de véhicule.
4. **Abstraction :**
    - Si votre langage le permet, utilisez une classe abstraite pour Véhicule.

➢ **Étape 2 – Méthodes et Modèles de Conception Orientée Objet**

Objectif : Organiser les relations logiques sans UML formel.

Travail demandé :

- Décrivez dans un document texte (ou commentaire dans le code) :
    - les relations entre vos classes (exemple : un Client peut avoir plusieurs ContratLocation),
    - les rôles et responsabilités de chaque classe,
    - les interactions principales (par exemple : création d'un contrat, calcul du tarif, retour d'un véhicule).

But : les étudiants apprennent à "penser" leur système avant de coder, même sans diagrammes UML formels.

➢ **Étape 3 – Langage de Programmation Orienté Objet**

Objectif : Implémenter le code.

Travail demandé :

- Implémentez toutes les classes et méthodes définies.
- Ajoutez un **programme principal (main.py)** qui permet :
    1. d'ajouter des clients,
    2. d'ajouter des véhicules,
    3. de créer un contrat de location,
    4. d'afficher la liste des contrats actifs,

5. de calculer le tarif total d'une location selon le type de véhicule **et la durée.**

Exemple : Tarif Voiture = 15 000 FCFA/jour, Tarif Moto = 8 000 FCFA/jour.

- Testez le polymorphisme avec une liste de Vehicule contenant des Voiture et Moto.

➢ Étape 4 – Environnements et Frameworks Orientés Objet

Objectif : Utiliser un environnement professionnel (IDE ou mini-framework).

Travail demandé :

1. Réalisez le projet dans un IDE (VS Code).
2. Pour les étudiants plus avancés :
    - utilisez un mini-framework selon le langage choisi :
        - Django (Python) : interface web simple de gestion des véhicules.
3. Intégrez un mécanisme simple d'entrée/sortie :
    - Sauvegarde des données dans un fichier texte ou JSON.
    - Lecture des véhicules enregistrés au démarrage du programme.

➢ Étape 5 – Gestion de Projet et Versionning

Objectif : Simuler un développement collaboratif.

Travail demandé :

1. Créez un dépôt Git local (git init) et effectuez vos commits à chaque étape.
2. Ajoutez un fichier README.md décrivant le projet et son fonctionnement.
3. Utilisez des branches pour développer des fonctionnalités séparément (par exemple : feature/contrat-location).
4. Montrez un historique clair des commits (git log --oneline --graph).

➢ Étape 6 – Cycle de Vie du Développement OO

Objectif : Boucler sur le processus complet.

Travail demandé :

1. **Analyse :** Listez dans un document .txt les besoins initiaux que votre logiciel couvre.
2. **Conception :** Expliquez brièvement les améliorations possibles.

3. **Implémentation :** Fournissez le code final.
4. **Test :** Montrez les sorties obtenues lors des tests.
5. **Maintenance :** Proposez deux fonctionnalités futures (ex : export PDF, gestion des pénalités de retard).