

Programmation C / Structure de données

mini projet de session

SMI4/SMA4

Le nom : ADIL

Le prenom : mohamed

CNE : 19-58220946

Date d'envoi : 6/3/2021

int chercher_voiture(int Num)

la fonction de type entier recu un seul argument de type entier. elle vérifie si le id de voiture est déjà existant si il existe elle retourner la valeur 1 sinon elle retourne -1

int chercher_contrat(float Num) et int chercher_client(int Num) travail avec la même manière

void Ajouter_voiture()

cette procédure permet d'ajouter des informations d'une nouvelle voiture

elle demande à l'utilisateur de saisir un id

dans la boucle while elle appelle la fonction **chercher_voiture()** pour vérifier si la voiture est déjà existante par comparaison la valeur id que l'utilisateur saisit avec les id dans le fichier **voiture.txt**

si la fonction **chercher_voiture()** retourne la valeur 1 alors la voiture existe donc la boucle va continuer jusqu'à l'utilisateur saisir un id valable

après la, l'utilisateur peut saisir d'autres informations et les enregistrer dans leur fichier

void Ajouter_client() travail avec la même manière

void Supprimer_voiture()

cette procédure crée un fichier temporaire **voiture_tmp.txt**

après demande à l'utilisateur de saisir l'id de voiture qu'il veut supprimer

dans la boucle, elle compare le id de cette voiture avec les informations dans le fichier **voiture.txt** ligne par ligne

et elle copie toutes les lignes qui ne contiennent pas cette id vers **voiture_tmp.txt**

après elle supprime **voiture.txt**

elle renomme **voiture_tmp.txt** par **voiture.txt**

void Ajouter_client() et **void Supprimer_contrat()** travail avec la même manière

void Modifier_voiture()

cette procédure crée un fichier temporaire **voiture_tmp.txt**

après demande à l'utilisateur de saisir l'id de voiture qu'il veut modifier

dans la boucle elle compare le id de cette voiture avec les informations dans le fichier **voiture.txt** ligne par ligne

et elle copie toutes les lignes vers **voiture_tmp.txt**

jusqu'à il arrive à l'id de voiture convenable

elle demande à l'utilisateur de saisir les nouvelles valeurs

et supprimer **voiture.txt**

enfin elle renomme **voiture_tmp.txt** par **voiture.txt**

void Modifier_contrat() et **void Modifier_client()** travail avec la même manière

void Lister_voiture()

cette procédure accède au fichier **voiture.txt** après elle lit le fichier ligne par ligne et les affiche

void Lister_client() travail avec la même manière

void voiture_menu()

l'interface de gestion des voitures qui contient des choix

et selon le numéro choisi, cette interface appelle la fonction respective

Lister_voiture() pour le nombre 1

Ajouter_voiture() pour le nombre 2

Modifier_voiture() pour le nombre 3

Supprimer_voiture() pour le nombre 4

main_menu() pour le nombre 5

la boucle **while(1)** continue jusqu'à l'utilisateur choisit de quitter le programme

void contrat_menu() et **void client_menu()** travail avec la même manière

pour la fonction **main_menu()** pour revenir à l'interface principale du programme

void main_menu()

L'interface générale du programme pendant la sélection appelle les fonctions **void voiture_menu()** et **contrat_menu()** et **client_menu()**

après l'avoir essayé

La boucle continue jusqu'à ce que l'utilisateur termine le programme

Void Modifier_location(int id)

cette fonction a un seul argument de type entier qui contient l'id de voiture

de même manière comme la procédure **Modifier_voiture()** sauf qu'elle change **EnLocation** seulement

si elle "non" reçoit "oui" et par l'inverse

void Louer_voitures()

elle vérifie que le numéro de contrat existe déjà

après la, elle vérifie la voiture

si la voiture n'existe pas elle retourne à la fenêtre précédente

et pour le client qui n'existe pas, elle retourne à la fenêtre de **gestion des clients**

et après de saisir des informations

cette procédure appelle la fonction **Modifier_location(idvoiture)** pour changer le statut de **EnLocation** de cette voiture

void Visualiser_contrat()

demande d'utilisation de saisir un nombre après elle cherche et compare les numéros de contrat dans le fichier

ContratsLocations.txt ligne par ligne

si il existe elle affiche toutes les informations dans cette ligne

sinon le contran n exist pas

void Retourner_voiture()

demande d'utilisation de saisir le id d'une voiture

si la voiture n'exist pas il retourn a la fentre Précédent

elle travali comme la fonction **Supprimer_contrat()** mais cette fois elle comparer le **ID** au lieu de **numero de facture**
apres, elle appel la **fonction Modifier_location(idvoiture)** pour changer le statut de **EnLocation**