

Mini projet python :

Instructions pour la Soumission du Projet :

Implémentation du Code :

- Implémentez le code en suivant les spécifications fournies dans l'énoncé du projet.
- Assurez-vous d'avoir correctement séparé le code de connexion à la base de données dans un fichier distinct database.py.

Utilisation de Git :

- Initialisez un dépôt Git local dans le répertoire de votre projet.
- Effectuez des commits réguliers pour enregistrer vos changements localement.
- Lorsque vous avez terminé une fonctionnalité ou une partie du projet, faites un push sur la branche (habituellement main ou master).
- Poussez votre code vers votre dépôt Git personnel sur la branche (habituellement main ou master)

Soumission sur Classroom :

- Accédez à la plateforme Classroom et ouvrez le projet correspondant.
- Dans la section de soumission, fournissez le lien vers votre dépôt Git personnel où se trouve votre code complet et fonctionnel.
- Assurez-vous que le dépôt Git est accessible en lecture par le professeur pour l'évaluation.

Vérification de la Soumission :

- Avant de soumettre, assurez-vous que tous les fichiers nécessaires (y compris database.py et les fichiers de classe Python) sont présents dans votre dépôt Git.
- Vérifiez que le lien fourni mène directement à votre dépôt Git et non à une page d'accueil ou à un répertoire spécifique.

Délai de Soumission :

- Respectez le délai de soumission indiqué sur la plateforme Classroom. Les soumissions tardives peuvent être pénalisées selon les règles du cours.

Partie 1:

Gestion d'une liste d'objets :

- Créez une classe ListePersonnes qui contient une liste d'objets de type Personne.
- Ajoutez une méthode ajouter_personne(nom, age) pour ajouter une nouvelle personne à la liste et enregistrer ces informations dans une table "Personnes" de la base de données MySQL.
- Ajoutez une méthode afficher_personnes() pour afficher les détails de toutes les personnes dans la liste en récupérant les données depuis la table "Personnes" de la base de données MySQL.

Recherche dans une liste d'objets :

- Ajoutez une méthode rechercher_personne(nom) à la classe ListePersonnes qui recherche une personne par son nom dans la table "Personnes" de la base de données MySQL et affiche ses détails si elle est trouvée.

Filtrage des personnes par âge :

- Ajoutez une méthode filtrer_personnes_par_age(min_age, max_age) à la classe ListePersonnes qui récupère les détails des personnes dont l'âge est compris entre min_age et max_age depuis la table "Personnes" de la base de données MySQL et les affiche.

Gestion d'une file d'attente :

- Créez une classe FileAttente pour gérer une file d'attente de personnes.
- Ajoutez une méthode ajouter_personne_en_attente(nom) pour ajouter une personne à la file d'attente et enregistrer son nom dans une table "FileAttente" de la base de données MySQL.
- Ajoutez une méthode supprimer_personne_de_attente() pour supprimer la première personne de la file d'attente en récupérant son nom depuis la table "FileAttente" de la base de données MySQL et l'afficher.

Priorisation dans la file d'attente :

- Modifiez la classe FileAttente pour qu'elle puisse gérer des personnes prioritaires.
- Ajoutez une méthode ajouter_personne_prioritaire(nom) pour ajouter une personne prioritaire à la file d'attente et enregistrer son nom dans la table "FileAttente" de la base de données MySQL.
- Modifiez la méthode supprimer_personne_de_attente() pour supprimer en priorité une personne prioritaire si elle existe dans la table "FileAttente" de la base de données MySQL, sinon supprimer la première personne normale.

Simulation d'un système de réservation :

- Créez une classe SalleCinema pour gérer les réservations dans une salle de cinéma.
- Ajoutez une méthode reserver_place(nom, place) pour réserver une place pour une personne et enregistrer cette réservation dans une table "Reservations" de la base de données MySQL.
- Ajoutez une méthode afficher_places_reservees() pour afficher les places réservées en récupérant les données depuis la table "Reservations" de la base de données MySQL.

Partie 2:

Gestion de la capacité de la salle :

- Ajoutez une méthode nombre_places_disponibles() à la classe SalleCinema pour afficher le nombre de places disponibles en consultant la table "Reservations" de la base de données MySQL et calculer les places disponibles.
- Ajoutez une vérification dans la méthode reserver_place(nom, place) pour s'assurer qu'il reste des places disponibles dans la salle en consultant la table "Reservations" de la base de données MySQL avant de réserver.

Filtrage des réservations par personne :

- Ajoutez une méthode filtrer_reservations_par_personne(nom) à la classe SalleCinema pour afficher les réservations faites par une personne spécifique

en récupérant les données depuis la table "Reservations" de la base de données MySQL et en filtrant par nom.

Annulation des réservations :

- Ajoutez une méthode annuler_reservation(nom) à la classe SalleCinema pour annuler toutes les réservations faites par une personne spécifique en supprimant les données correspondantes dans la table "Reservations" de la base de données MySQL.

Gestion des places spéciales :

- Modifiez la classe SalleCinema pour qu'elle puisse gérer des places spéciales pour les personnes handicapées.
- Ajoutez une méthode reserver_place_speciale(nom) pour réserver une place spéciale pour une personne handicapée et enregistrer cette réservation dans la table "Reservations" de la base de données MySQL.