

# Rapport SOAP

*Version Centralisée*



## Organisation de la solution :

Pour réaliser l'objectif de ce TP SOAP qui est la création d'une application de réservation d'hôtel en ligne, j'ai créé **quatre classes** et **une classe Main** qui sert à exécuter le programme.

### **1. Classe Hôtel :**

Dans cette classe, j'ai déclaré des attributs pour décrire un hôtel (comme son nom, son adresse, son nombre d'étoiles), conformément au diagramme UML. J'ai ensuite créé un constructeur et les accesseurs (getters et setters) pour ces attributs.

J'ai ajouté par la suite des méthodes importantes :

- **Méthode ajoutChambre :**

Cette méthode sert à ajouter des chambres à l'hôtel. Elle initialise une liste de chambres disponibles pour chaque hôtel.

- **Méthode rechercheDisponibilites :**

Cette méthode permet de rechercher les chambres disponibles dans un hôtel en fonction des critères spécifiés par l'utilisateur (comme la date d'arrivée, la date de départ, le nombre de personnes, et le prix maximum).

Elle parcourt la liste des chambres de l'hôtel et retourne uniquement celles qui répondent à ces critères.

### **Classe Chambre :**

Dans cette classe, j'ai déclaré des attributs pour décrire une chambre (comme son type, son numéro, son prix, sa capacité...). J'ai ensuite créé un constructeur ainsi que des getters et setters (accesseurs) pour manipuler ces attributs.

Ensuite, j'ai ajouté une :

### **Méthode ajoutReservation :**

Cette méthode permet d'ajouter une réservation à la liste des réservations pour cette chambre.

Elle stocke les réservations associées à cette chambre dans une liste, ce qui permet de vérifier les disponibilités de la chambre pour des dates données.

### **Méthode disponibleSurPeriode :**

Cette méthode vérifie si la chambre est disponible pendant une période spécifique (définie par une date d'arrivée et une date de départ).

Elle parcourt la liste des réservations existantes de la chambre et vérifie si les dates demandées se chevauchent avec les dates des réservations existantes.

**Pour la classe Réservation**, j'ai uniquement déclaré les différents attributs correspondant aux informations nécessaires pour décrire une réservation, comme le numéro de réservation, les dates, le prix, et les relations avec l'hôtel, la chambre, et le client.

Concernant la méthode **effectueReservation** :

Cette méthode vérifie si la chambre sélectionnée est vide pour une période précise. Si c'est le cas, cela signifie que la chambre est disponible.

**La classe Client** : j'ai créé une structure simple pour représenter un client. J'ai déclaré les attributs (numCin, nom, prénom, email et le numéro de téléphone). J'ai également conservé la même structure cohérente pour la création du constructeur et des accesseurs (getters et setters), afin de permettre une gestion organisée et uniforme des données.

### **Pour la classe Main :**

#### 1- La création des instances des objets :

J'ai créé une instance de la classe Hôtel pour représenter l'hôtel dans le système, en fournissant les mêmes informations passées en paramètres dans la classe Hôtel initiale.

J'ai également créé quatre instances de la classe Chambre pour décrire les chambres disponibles dans cet hôtel, chacune avec ses paramètres.

#### 2- Ajout des chambres à l'hôtel :

J'ai utilisé la méthode **ajoutChambre()** pour associer chaque chambre créée à l'hôtel concerné. Cela permet d'enregistrer les chambres dans une liste qui va être utilisée plus tard pour la recherche de disponibilité.

#### 3- Recherche des chambres disponibles :

J'ai défini des critères de recherche pour simuler une recherche d'hôtel. (comme la date d'arrivée, la date de départ, le nombre de personnes, et le prix maximum que le client ne compte pas dépasser.

Grâce à la méthode **rechercheDisponibilites()** de la classe Hôtel, seules les chambres répondant aux critères sont affichées. Pour cela j'ai opté pour la méthode isEmpty dans la condition **if chambresDisponibles.isEmpty** qui vérifie si la liste des chambres disponibles

est vide. Une fois la liste est vide cela signifie qu'il n'y a aucune chambre correspondant aux critères de recherche de l'utilisateur.

#### 4. Interaction avec l'utilisateur via le Scanner :

Une fois la liste des chambres disponibles sera affichée, l'utilisateur pourra effectuer son choix par toute flexibilité pour cela j'ai intégré l'objet Scanner pour permettre à l'utilisateur de faire son choix parmi les résultats affichés. Cette interaction rend le programme dynamique et engageant.

- L'utilisateur peut entrer le numéro correspondant à la chambre qu'il souhaite réserver.
- fois que l'utilisateur a choisi une chambre, j'ai ajouté une étape pour lui demander ses coordonnées bancaires. Une fois les coordonnées sont fournies, une vérification doit être établie pour garantir la validité de ces coordonnées (pas nulles ou vides) avant de finaliser la réservation.

#### 5. Validation et confirmation de la réservation :

- Si toutes les conditions sont réunies, la réservation est confirmée, et un message avec le numéro de réservation sera affiché.