

Compte Rendu – Mini-Projet

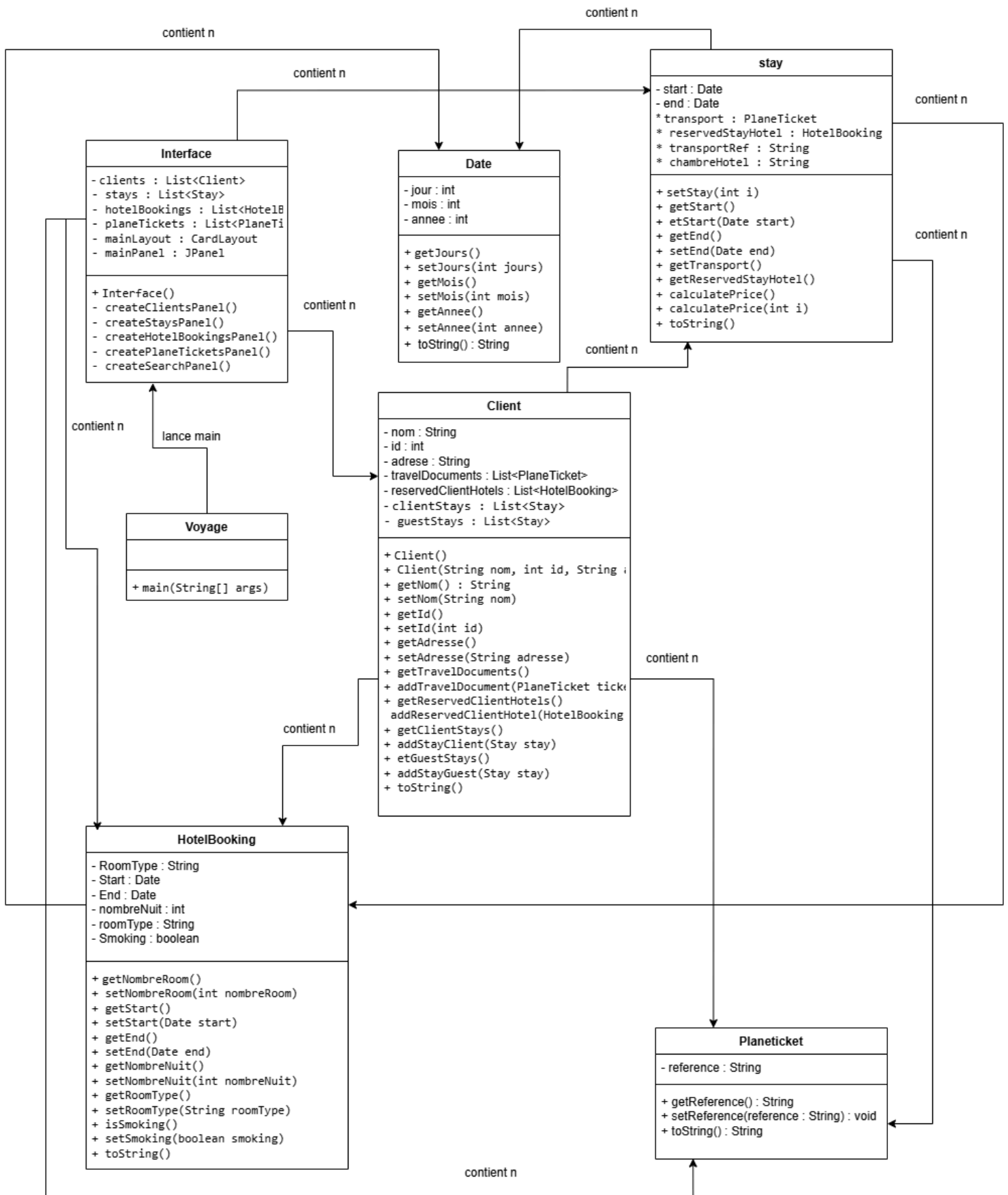
Java : Gestion de Voyages

I\ But du Projet :

L'objectif du projet était de développer une **application de gestion de voyages** en **Java**, en utilisant la **Programmation Orienté Objet** et pour l'apparence l' **interface graphique Swing**, permettant les fonctionnalités suivantes :

- Gestion des clients :
 - Ajouter un client
 - Visualiser les détails d'un client
 - Associer un client à un séjour ou en tant qu'invité
- Gestion des séjours :
 - Ajouter un nouveau séjour avec date de début/fin
 - Ajouter un transport (billet d'avion)
 - Ajouter une réservation d'hôtel
- Réservations d'hôtel :
 - Spécifier le type de chambre (Single, Double, Family)
 - Définir le nombre de nuits, dates, fumeur ou non
- Billets d'avion :
 - Ajouter un billet avec référence
- Recherche et affichage :
 - Rechercher les séjours d'un client
 - Afficher toutes les réservations
 - Calculer le prix d'un séjour (méthode calculatePrice())
- Interface graphique :
 - Interface Java Swing
 - Menu principal avec navigation simple
 - Formulaire de saisie et tableaux d'affichage

II\ Diagramme de Classes :



III\ Brève explication du fonctionnement :

Le projet simule une **agence de voyage virtuelle**. Chaque client peut être relié à un ou plusieurs séjours, chacun contenant un billet d'avion et une réservation d'hôtel. Le système permet d'ajouter, afficher et rechercher les séjours, et **calculer automatiquement le prix total d'un séjour**. La structure repose sur des **relations bien définies entre objets** (composition, association), une interface Swing pour l'ergonomie, et une interface Interface pour la persistance des données.

IV\ Détail des Classes :

1) Classe Client :

Représente un client de l'agence.

Attribut(s) : nom, id, adresse, travelDocuments, reservedClientHotels, clientStays, guestStays

Méthode(s) : les 2 constructeurs de la classe (un sans paramètre et un avec le nom, id et l'adresse), les getters/setters/adders des différents attributs (pas pour tout les attributs) de la classe Client

2) Classe Date :

Représente une date (jours/mois/annee)

Attribut(s) : jours, mois, annee

Méthode(s) : le constructeur de la classe prenant en paramètre jours, mois, annee, les getters/setters de tous les attributs et toString()

3) Classe HotelBooking :

Représente la réservation d'une chambre d'hôtel lors d'un séjour

Attribut(s) : nombreRoom, Start, End, nombreNuit, roomType, Smoking

Méthode(s) : les 2 constructeurs de la classe (un sans parametre et un avec tous les attributs de la classe), les getters/setters des attributs, isSmoking(), setSmoking() et toString()

4) Classe Interface :

Gère toute l'interface graphique du projet et hérite de JFrame

Attribut(s) : mainLayout, mainPanel, clients, stays, hotelBookings, planeTickets

Méthode(s) : le constructeur de la classe sans attribut, createClientsPanel(), createStaysPanel(), createHotelBookingsPanel(), createPlaneTicketsPanel(), createSearchPanel()

5) Classe PlaneTicket :

Représente le billet d'avion d'un séjour

Attribut(s) : Reference

Méthode(s) : les getter/setter de Reference et toString.

6) Classe Stay :

Représente un séjour de voyage, elle regroupe toutes les informations des voyages des clients

Attribut(s) : Start, End, transport, reservedStayHotel, transportRef, chambreHotel

Méthode(s) : les gettters/setters de la plupart des attributs, 2 calculatePrice() (un sans paramètre et un avec un entier) et toString()

7) Classe Voyage :

Classe main qui lance tout le code

Attribut(s) : Aucun attribut

Méthode(s) : main()