

# II.1102 Algorithmique et Programmation JAVA

## - Projet Assisté : 1<sup>ÈRE</sup> Séance -

### 1. Informations générales

Le projet portera sur le développement d'une application qui gère un système de réservation d'une compagnie aérienne avec toutes les fonctionnalités requises. Plus d'informations sur ce sujet en section 2 et section 3. Avant de décrire les besoins de ce projet, voici quelques consignes générales :

1. Le projet se fait en binôme (éventuellement individuel).
2. Il est absolument interdit d'utiliser les outils de la génération du code. Ceci sera considéré comme un plagiat flagrant.
3. Des bonus pourront être attribués pour des fonctionnalités originales de votre choix.

### 2. Contraintes techniques

- L'application doit être développée en Java en utilisant des concepts de programmation orientée objet,
- Utilisation de structures de données appropriées pour le stockage des informations (tableaux, listes, etc.),
- Travailler avec un service d'hébergement de projets en cours de développement : **GitHub**
- Utiliser un outil de gestion et d'automatisation de production des projets logiciels Java: **Maven**

### 3. Liens utiles :

- Tutoriel d'utilisation de Github avec IntelliJ Idea :  
<https://www.jetbrains.com/help/idea/github.html>

- Comment configurer un projet avec GitHub :

<https://www.jetbrains.com/help/idea/manage-projects-hosted-on-github.html>

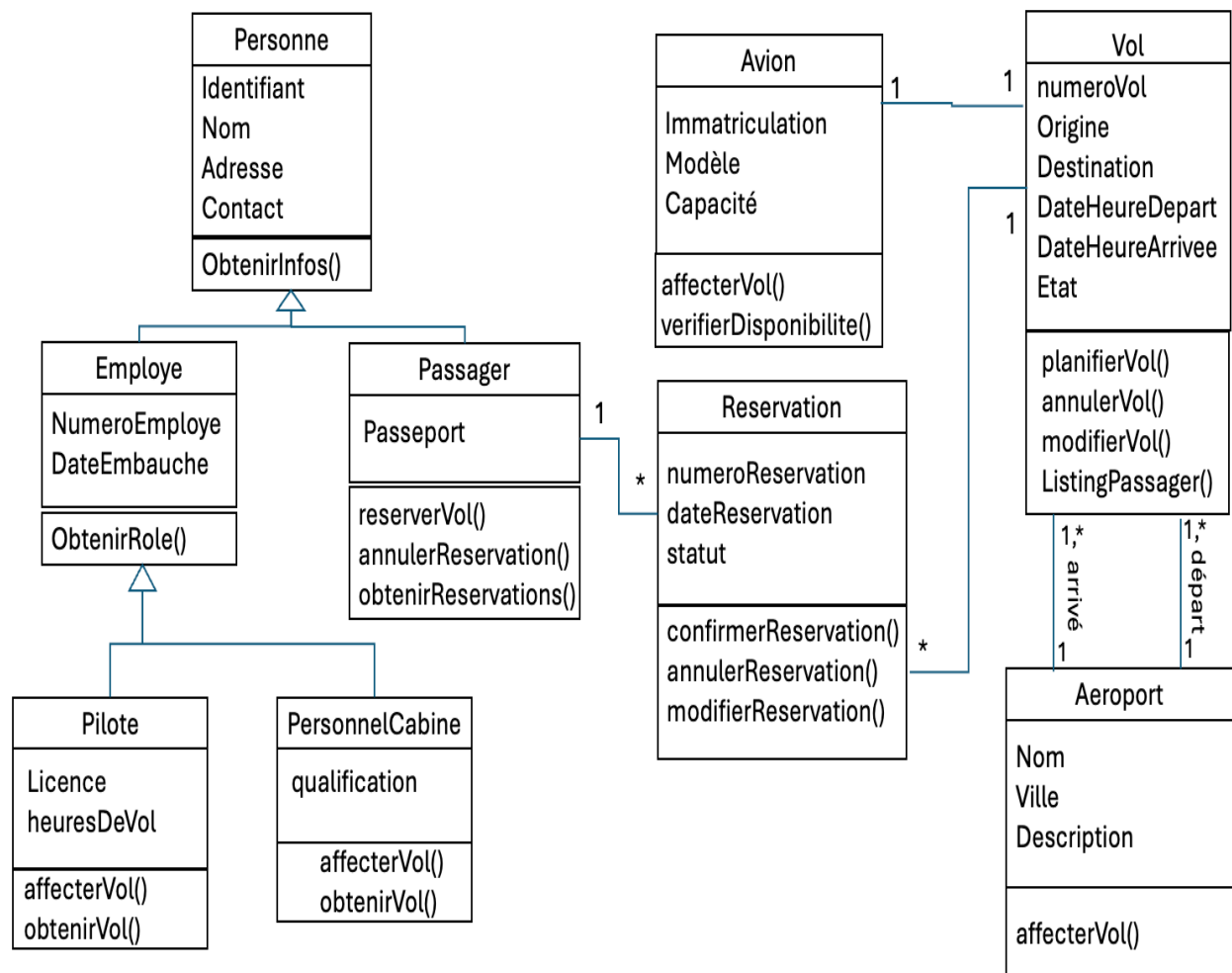
- Tutoriel d'utilisation de Maven sur IntelliJ Idea :  
<https://www.jetbrains.com/help/idea/maven-support.html>

- Vidéo d'installation maven sur IntelliJ Idea :

<https://www.jetbrains.com/guide/java/tutorials/marco-codes-maven/installing-maven/>

### 4. Présentation du projet : Application de gestion d'une compagnie aérienne

Un système de réservation de compagnies aériennes basé sur Java qui utilise la programmation orientée objet. Le système peut gérer les clients, l'équipage, la réservation et l'annulation de vols. Vous allez implémenter les différentes classes issues du diagramme illustré sur la figure ci-dessous à l'aide de concepts POO tels que l'héritage, l'encapsulation, l'association et la composition.



## 5. Fonctionnalités du projet

En se basant sur le diagramme ci-dessus, vous allez créer toutes les classes tout en considérant les contraintes relationnelles schématisées (héritage, associations, etc.)

Dans chaque classe, vous allez ajouter les méthodes CRUD qui permettent d'ajouter, modifier, chercher et supprimer les instances d'objets.

Pour chaque classe créée, il faut implémenter les différentes méthodes mentionnées sur le diagramme. Les principales méthodes à implémenter sont ci-suit :

- **Gestion des Personnes :**

- **obtenirInfos()** : c'est une méthode qui permet d'afficher les différentes informations d'une personne .
- **obtenirRole()** : cette méthode retourne le rôle d'un employé par son identifiant (pilote, personnel cabine, etc.).
- **Gestion des réservations :**
  - **reserverVol()** : une méthode déclenchée par la classe 'Passager'. Un passager peut réserver plusieurs vols. Une liste de vols réservés doit être ajoutée dans la classe réservation.
  - **annulerReservation()** : Cette méthode permet d'annuler une réservation donnée par son ID (numéroRéservation).
  - **obtenirReservations()** : Cette méthode permet d'obtenir la liste d'informations sur une réservation donnée par son ID (numéroRéservation).
- **Gestion des vols :**
  - **affecterVol()** : Un équipage doit être affecté à chaque vol. Un pilote ainsi qu'une équipe cabine devra assurer l'acheminement d'un vol.
  - **obtenirVol()** : Cette méthode permet d'obtenir la liste d'informations sur un vol donné par son ID (numéroVol).
  - **planifierVol()** : l'activité de la compagnie doit être planifiée. Un ensemble de vol sera ajouté sur une période donnée (un planning sur une journée par exemple)
  - **annulerVol()** : Cette méthode permet d'annuler un vol donné par son ID (numéroVol).
- **Gestion des avions :**
  - **affecterVol()** : Pour chaque vol, on doit affecter un appareil de la liste des avions disponibles.
  - **VerifierDisponibilite()** : Avant l'affectation d'un avion à un vol donné, il faut vérifier sa disponibilité dans l'horaire prévu.

## 6. Fonctionnalités avancées (Bonus)

- **Statistiques et rapports :**
  - Génération de rapports sur le nombre de vols, passagers transportés, revenus générés.
  - Affichage des destinations les plus populaires.

## 7. Livrables

Les livrables à déposer sur Moodle avant la date limite :

- **Un dossier compressé** contenant tout votre code rédigé (Date limite : 5 décembre).