

Projet Java

Notes en préambule :

- le travail doit se faire en binome ;
- soyez concis et précis, et justifiez vos réponses par des commentaires appropriés ;
- soyez rigoureux dans la syntaxe Java ;

Étude de cas. L'équipage d'un avion est toujours constitué d'un pilote, d'un copilote et de plusieurs personnels naviguants et commerciaux (PNC). Chacune de ces personnes est identifiée par son nom et sa fonction. L'équipage d'un avion est reconstitué pour chacun de ses vols.

Chaque membre d'équipage doit être opérationnel sur deux types d'avions (par exemple le PNC Richard est opérationnel sur Airbus A320, et Boeing 747). Chaque type d'avion a un nombre de PNC dans son équipage oscillant entre un minimum et un maximum. Pour l'exemple nous choisirons le nombre de PNC des Airbus A320 entre deux et trois et ceux d'un B747 entre trois et quatre.

L'objectif du système à modéliser est de constituer des tableaux de service. Les membres de l'équipage peuvent visualiser les vols dans lesquels ils ont été affectés. L'administrateur du système peut créer et supprimer des entités dans le système. Le manager peut ajouter et supprimer des personnes dans un équipage pour un vol donné (un vol est désigné par un numéro de vol et une date), les données du vol sont archivées après le vol pour une année.

La figure 1 est un extrait du tableau de service de quelques employés de la compagnie AIR France. Pour simplifier, le tableau n'en représente que quelques-uns.

Avion	VOL	Dest	Date	Type	Site	Prénom	Nom	Fonction
13562	AF347	Londres	11/10/06	A320	Orly	Corinne	Lancel	Pilote
13562	AF347	Londres	11/10/06	A320	Orly	Amy	Bosch	Copilote
13562	AF347	Londres	11/10/06	A320	Orly	Maureen	Gates	PNC
13562	AF347	Londres	11/10/06	A320	Orly	Richard	Tata	PNC
13562	AF347	Londres	11/10/06	A320	Orly	Ben	Gamma	PNC
32156	AF545	New-York	12/10/06	B747	Roissy	Jacques	Vlisside	Pilote
32156	AF545	New-York	12/10/06	B747	Roissy	Jean-Louis	Dupont	Copilote
32156	AF545	New-York	12/10/06	B747	Roissy	Ségolène	Orval	PNC
32156	AF545	New-York	12/10/06	B747	Roissy	François	Saadi	PNC
32156	AF545	New-York	12/10/06	B747	Roissy	Nicolas	Harper	PNC

FIGURE 1 – Extrait du tableau de service des vols AF347 et AF545 sur une période

Les classes et -attributs- obtenus après analyse du texte sont les suivants :

- TableauVols -- la classe façade d'accès au système ;

- Équipage -auMin (booléen min atteint), auMax (booléen max atteint)- la classe équipage regroupe l'ensemble des membres d'équipage (pilote, copilote et PNC) qui participent à un vol (à une date donnée) ;
- Avion -- un avion a un type donné, il participe à plusieurs vols ;
- Pilote -- un membre d'équipage qui a pour métier pilote ;
- Copilote -- un membre d'équipage qui a pour métier copilote ;
- PNC -- un membre d'équipage qui a pour métier PNC;
- MembreÉquipage -nom, prénom-
- TypeAvion -nom, nbPNCmin,nbPNCmax- un membre d'équipage ne peut voler que sur deux types d'avion. Plusieurs avions peuvent être du même type, d'où la nécessité de créer une classe TypeAvion ;
- Vol -numéro, site, destination, date- un vol est une classe d'association entre un avion et un ensemble de membres d'équipage. Un vol a lieu a une date donnée.

La figure 2 présente les acteurs et les cas d'utilisation du système. La figure 3 présente le diagramme de classes proposé.

Les prototypes des constructeurs de ces classes sont les suivants :

```
Avion(TypeAvion type, String ref)
CoPilote(String nom, String prenom) throws InvariantBroken
Equipage(Vol v)
Personne(String nom, String prenom) throws InvariantBroken
Pilote(String nom, String prenom) throws InvariantBroken
PNC(String nom, String prenom) throws InvariantBroken
TableauVols(String nom)
TypeAvion(String nom, int min, int max)
TypeAvion(String nom)
Vol(String num, String site, String dest, Avion av, Date dep)
Vol(String num, Date dep)
```

Voici quelques méthodes parmi les autres méthodes des classes du système :

Classe Avion :

```
public String getReference()
public TypeAvion getType()
```

Classe Personne :

```
public String getNom()
public String getPrenom()
public boolean peutVoler(TypeAvion t)
public boolean addQualification(TypeAvion type) throws EquipageException, InvariantBroken
public boolean delQualification(TypeAvion type, boolean fromType)
```

Classe Vol :

```
public void addPilote(Pilote p) throws EquipageException
public void addCoPilote(CoPilote c) throws EquipageException
public boolean addPNC(PNC p) throws EquipageException
public boolean equipageAuComplet()
```

Classe Equipage :

```
public void addPilote(Pilote p) throws EquipageException
public void addCoPilote(CoPilote c) throws EquipageException
public boolean addPNC(PNC p) throws EquipageException
Classe TypeAvion :
public void addQualifie(Personne m) throws EquipageException
public void delQualifie(Personne m) throws EquipageException
public void purgeQualifies() throws EquipageException, InvariantBroken
public String getNom()
public int getMinPNC()
public int getMaxPNC()
```

Partie 1 : Conception

1. Reprendre la conception proposée en y utilisant et en justifiant :
 - Des packages, Des interfaces
 - Des classes abstraites, internes
2. Proposer au moins deux exceptions et utiliser les

Partie 2 : Persistance

1. L'ensemble des aéroports est stockés dans un fichier donnée en sérialisant la classe correspondante
2. L'ensemble des données des vols, personnels est stocké dans une base

Partie 3 : Interface Graphique

1. Proposer une interface graphique pour afficher le tableau
2. Proposer une interface graphique pour mettre à jour les données de vols par un des acteurs

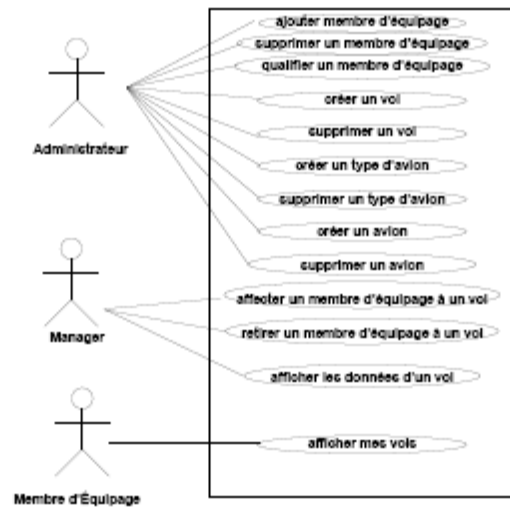


FIGURE 2 – Diagramme de cas d'utilisation

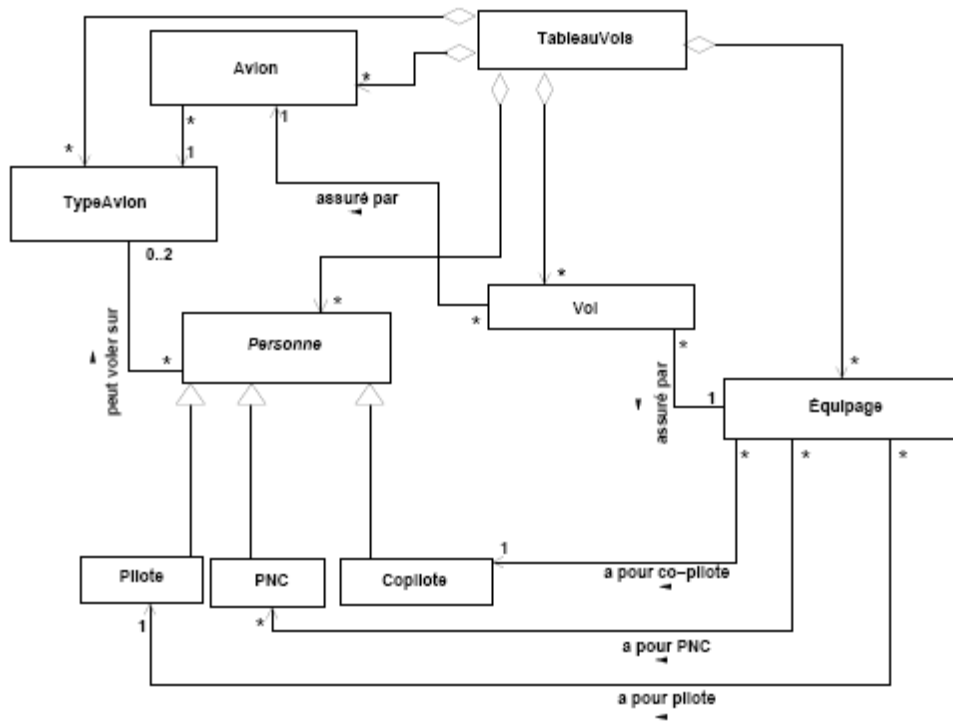


FIGURE 3 – Diagramme de classes