Université Paris Ouest Nanterre 2015-2016

Master 1 MIAGE SEGMI

POO L.M. Hillah, P. Poizat

A irline M anagement System

Cet ´enonc´e comporte 4 pages.

1. Objectif

L’objectif decedevoir est demettreen oeuvrelesbonnespratiqueset patronsdeconceptionsen program-

mation orient´ee-objet. Il s’agit de r´ealiser une application de gestion de syst`eme de gestion a´eroportuaire,  
dont le comportement global est r´eparti dans un ensemble de classes, pr´esent´ees dans la ﬁgure 1.

L’application doit permettre `a un client de cr´eer des a´eroports, des compagnies a´eriennes ainsi que des

vols. Le point d’entr´ee de l’application est le Syst emManager . Chaque compagnie Ai r l i ne est associ´e `a un  
ensembledevols(Fl i ght s). Un vol aun a´eroport ded´epart (or i gi n) et un a´eroport d’arriv´ee(dest i nat i on).  
L’origine et la destination ne peuvent ˆetre les mˆemes.

Chaque vol comprend des classes tarifaires (e.g., premi`ere classe, business) appel´ees f l i ght sect i on.

Chaque classe tarifaire comprend des si`eges organis´es en rangs (ligne et colonne).

1.1. Consigne

Vousdevez fairefonctionner votreapplication sur lejeu minimal d’instructionsfourni dansleprogramme

principal `a la ﬁn de ce document. Vous devez ´egalement utiliser un framework de test (JUnit, TestNG),  
pour fournir des jeux de test pour les fonctions critiques de l’application que vous identiﬁerez et dont vous  
justiﬁerez le caract`ere critique (pas de tests sur les getters/setters!).

2. SystemM anager

C’est le point d’entr´ee de l’application. Les clients interagissent avec l’application en appelant les op´era-

tions oﬀ ertes par Syst emManager . Ce dernier est reli´e aux a´eroports et compagnies a´eriennes dans l’appli-  
cation. A sa cr´eation, le Syst emManager ne poss`ede aucun a´eroport ou compagnie a´erienne. Pour les cr´eer  
les op´erations cr eat eAi r por t () et cr eat eAi r l i ne() doivent ˆetre invoqu´ees.

Le Syst emManager contient ´egalement lesop´erationspour cr´eer lesclassestarifaires, trouver lesvolsdis-

ponibles entre deux a´eroports, et r´eserver des si`eges sur un vol. Pour aﬃ cher toute l’information concernant  
les a´eroports les compagnies et les vols, classes tarifaires et si`eges, on invoque l’op´eration di spl aySyst em-  
Det ai l s().

— cr eat eAi r por t (St r i ng n) : cr´ee un objet de type Ai r por t et le lie au Syst emManager . L’a´eroport

doit avoir un code n, dont la longueur est exactement ´egale `a 3. Deux a´eroports diﬀ ´erents ne peuvent  
avoir le mˆeme code.

1

name doit avoir au

plus 5 caractères

Airline

- name: String SystemManager

+createFlight: Flight

+ﬁndFlight: Flight \* 1 +createAirport()

+createSection() +createAirline()

+getAvailableFlights(): airlines +createFlight()

ArrayList<Flight> +createSection()  
+bookFlight() +ﬁndAvailableFlights()

airline 1 +bookSeat()

+displaySystemDetails()

ﬂights \* 1

Flight

- ﬂightDate: Calendar

- ﬂightID: String

\*

+getAirline(): Airline

+getID(): String

+getOrig(): Airport

+getDest(): Airport

1

destination

airports

Airport

- code: String

\*

+getDate(): Calendar 1 +getName(): String

+hasSection():Boolean

+hasSeats(): Boolean

\*

origin

code doit avoir

+createSection(): Boolean exactement 3   
+ﬁndSection(): FlightSection caractères

+bookSeat()

1 ﬂightID doit avoir au

plus 5 caractères

sections \*

FlightSection Seat

- section: SeatClass 1 1..\* - seatNum: SeatID

+getSeatClass: SeatClass

+hasAvailableSeats:

seats

- isBooked: Boolean

+getSeatNum(): SeatID

Boolean

+bookSeat(): Boolean

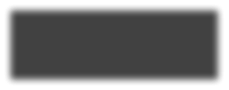
+getStatus(): Boolean

«enumeration»

Image_63_0

SeatClass SeatID

+ FIRST ("First") - row: Integer



+ BUSI ("Business") - column: Char

+ ECO ("Economy")  
- name: String

Figur e 1 – Description architecturale de Airline Management System

— cr eat eAi r l i ne(St r i ng n) : cr´eeunecompagniea´erienneet la lieau Syst emManager . Lenom n d’une

compagnie doit ˆetre de longueur au plus ´egale `a 5. Deux compagnies diﬀ ´erentes ne peuvent avoir le  
mˆeme nom.

— cr eat eFl i ght (St r i ng n, St r i ng or i g, St r i ng dest , i nt year , i nt mont h, i nt day, St r i ng

i d) : cr´eeun vol pour unecompagnien, en invoquant l’op´eration cr eat eFl i ght sur la classe Ai r l i ne.

— cr eat eSect i on(St r i ng ai r , St r i ng f l I D, i nt r ows, i nt col s, Seat Cl ass s) : cr´ee une sec-

tion tarifaire, de classe s, pour un vol identiﬁ´e par ﬂID, associ´e `a une compagnie air, en invoquant  
l’op´eration cr eat eSect i on() de la classe Ai r l i ne. La section contiendra le nombre de lignes et de  
colonnes.

— f i ndAvai l abl eFl i ght s(St r i ng or i g, St r i n dest ) : trouve tous les vols pour lesquels il existe

encore des si`eges disponibles, entre les a´eroports de d´epart et d’arriv´ee.

— bookSeat (St r i ng ai r , St r i ng f l , Seat Cl ass s, i nt r ow, char col ) : r´eserve le si`ege dont la

2

position est indiqu´ee par row et col (e.g. 15A), sur le vol ﬂ de la compagnie air.

— di spl aySyst emDet ai l s() : aﬃ chetoutel’information concernant touslesobjets(e.g. a´eroports, com-

pagnies, vols, si`eges, ...) dans le syst`eme.

3. A irport

Un objet de cette classe repr´esente un a´eroport. Il poss`ede un nom, de longueur 3 caract`eres.

4. A irline

Cette classe d´eﬁnit une compagnie a´erienne. Une compagnie poss`ede z´ero ou plusieurs vols en cours. A

la cr´eation d’un objet decetype, il n’y a initialement aucun vol. Chaquevol doit avoir un identiﬁant unique.

— Fl i ght cr eat eFl i ght (St r i ng or i g, St r i ng dest , Cal endar dat e, St r i ng i d) :cr´eeun vol pour

une compagnie a´erienne.

— Fl i ght f i ndFl i ght (St r i ng n) : trouve un vol dont l’identiﬁant est n.

— cr eat eSect i on(St r i ng f l I D, i nt r ows, i nt col s, Seat Cl ass s) : cr´ee une section tarifaire de

classes, pour un vol dont l’identiﬁant est ﬂID. La section contiendra lenombredeligneset decolonnes.

— Ar r ayLi st <Fl i ght > get Avai l abl eFl i ght s(Ai r por t or i g, Ai r por t dest ) : trouve tous les vols

sur lesquels il existe encore des si`eges disponibles, entre les a´eroports de d´epart et d’arriv´ee.

— bookFl i ght (St r i ng f l , Seat Cl ass s, i nt r ow, char col ) : r´eserve un si`ege dont la position est

indiqu´ee par row et col (e.g. 15A) dans la section tarifaire s, sur le vol ﬂ.

5. FlightSection

Cetteclassed´eﬁnit uneclasse(ou section) tarifaire. Chaquesection poss`edeuneclasse(premi`ere, aﬀ aires,

ou ´economique) et au moins 1 si`ege. Une FlightSection poss`ede des attributs nombre de rows et nombre de  
columns, aﬁn de savoir combien de si`eges elle contient et le calcul du nombre de si`eges disponibles.

Une section tarifaire contient au plus 100 rang´ees de si`eges et au plus 10 si`eges par rang´ee.

— hasAvai l abl eSeat s() renvoie vrai si et seulement si la section poss`ede encore des si`eges disponibles

(non r´eserv´es).

— bookSeat () r´eserver le premier si`ege disponible. Son utilisation est conditionn´ee `a celle de hasAvai -

l abl eSeat s().

— bool ean bookSeat (Seat I D sI d) r´eserver le si`ege `a l’emplacement d´esign´e par le param`etre sID, si

ce si`ege est disponible.

6. Seat

Cette classe d´eﬁnit un si`ege. Un si`ege poss`ede un identiﬁcateur, qui indique sa rang´ee et sa colonne

(caract`ere allant de A `a J). Il poss`ede ´egalement un statut qui indique s’il est r´eserv´e ou pas.

3

. Ce client appelle des

Vous ˆetes invit´es `a ´etendre cette classe client avec d’autres invocations pour tester le comportement

**123**

) ;

) ;

, 1, 1, Seat Cl ass. busi ness) ;

, 1, 2, Seat Cl ass. economy ) ;

, 5, 5, Seat Cl ass. economy ) ;

1  
2  
3  
4  
5  
6  
7  
8  
9

10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

29  
30  
31  
32  
33  
34  
35  
36

37  
38  
39  
40  
41  
42  
43  
44  
45

46  
47  
48  
49  
50  
51

7. Client de l’application

Un exemple de client de cette application est fourni dans la classe Cl i ent AMS

op´erations de la classe Syst emManager .

attendu de votre application.

**public class** Cl i ent AMS {

**public s t at i c void** mai n ( St r i ng[] ar gs) {

Syst emManager r es = **new** Syst emManager ( ) ;

/ / A i r por t s  
r es. cr eat eAi r por t (**"DEN"**) ;   
r es. cr eat eAi r por t (**"DFW"**) ;   
r es. cr eat eAi r por t (**"LON"**) ;   
r es. cr eat eAi r por t (**"DEN"**) ;   
r es. cr eat eAi r por t (**"CDG"**) ;   
r es. cr eat eAi r por t (**"JPN"**) ;   
r es. cr eat eAi r por t (**"DEN"**) ; / / Pb d ’ u ni ci t e  
r es. cr eat eAi r por t (**"DE"**) ; / / I nv al i de  
r es. cr eat eAi r por t (**"DEH"**) ;   
r es. cr eat eAi r por t (**"DRIrdn3"**) ; / / I nv al i de

/ / A i r l i nes  
r es. cr eat eAi r l i ne(**"DELTA"**) ;   
r es. cr eat eAi r l i ne(**"AIRFR"**) ;   
r es. cr eat eAi r l i ne(**"AMER"**) ;   
r es. cr eat eAi r l i ne(**"JET"**) ;   
r es. cr eat eAi r l i ne(**"DELTA"**) ;   
r es. cr eat eAi r l i ne(**"SWEST"**) ;   
r es. cr eat eAi r l i ne(**"FRONTIER"**) ; / / I nv al i de

/ / F l i ght s  
r es. cr eat eFl i ght (**"DELTA"**,  **"DEN"**,  **"LON"**, 2008, 11, 12,  **"**  
r es. cr eat eFl i ght (**"DELTA"**,  **"DEN"**,  **"DEH"**, 2009, 8, 9,  **"567"**  
r es. cr eat eFl i ght (**"DELTA"**,  **"DEN"**,  **"NCE"**, 2010, 9, 8,  **"567"**

I nv al i de

/ / Sect i ons  
r es. cr eat eSect i on(**"JET"**,  **"123"**, 2, 2, Seat Cl ass. economy ) ;   
r es. cr eat eSect i on(**"JET"**,  **"123"**, 1, 3, Seat Cl ass. economy ) ;   
r es. cr eat eSect i on(**"JET"**,  **"123"**, 2, 3, Seat Cl ass. f i r st ) ;   
r es. cr eat eSect i on(**"DELTA"**,  **"123"**  
r es. cr eat eSect i on(**"DELTA"**,  **"123"**  
r es. cr eat eSect i on(**"SWSERTT"**,  **"123"**

I nv al i de

r es. di spl aySyst emDet ai l s ( ) ;

r es. f i ndAvai l abl eFl i ght s(**"DEN"**,  **"LON"**) ;

r es. bookSeat (**"DELTA"**,  **"123"**, Seat Cl ass. busi ness , 1,  **’A ’** ) ;   
r es. bookSeat (**"DELTA"**,  **"123"**, Seat Cl ass. economy , 1,  **’A ’** ) ;   
r es. bookSeat (**"DELTA"**,  **"123"**, Seat Cl ass. economy , 1,  **’B ’** ) ;   
r es. bookSeat (**"DELTA"**,  **"123"**, Seat Cl ass. busi ness , 1,  **’A ’** ) ;

Deja r eser v e

r es. di spl aySyst emDet ai l s ( ) ;

r es. f i ndAvai l abl eFl i ght s(**"DEN"**,  **"LON"**) ;

}

}

4

**"**) ;

/ /

/ /

/ /