



PROGRAMACIÓ 1

CURS: 2023/2024-Q2

ACTIVITAT 1

1 Disseny modular en C++: Gestió de reserves de vols d'una aerolínia

La nova aerolínia de baix cost “Voling” ens ha demanat de col·laborar en un projecte de prova de concepte (o PoC, de l'anglès *Proof of Concept*) pel seu sistema de reserves de vols. Per aquesta prova, ens ha definit el conjunt de requeriments funcionals següent:

- Vols: Cal emmagatzemar i poder consultar en tot moment la informació sobre cada vol, com ara el número de vol, la ciutat d'origen, la ciutat destinació, l'hora de sortida, l'hora d'arribada, l'estat del vol i el nombre de seients disponibles.
- Passatgers: Cal mantenir la informació sobre cada passatger/a, com ara el nom, l'estat de la seva reserva i el número de seient assignat.
- Reserves: Cal emmagatzemar i poder consultar en tot moment la informació sobre cada reserva, com ara el número de reserva, el/la passatger/a, el número de vol i el mètode de pagament.

Per representar aquesta informació i crear el sistema de gestió de reserves de vols, farem servir 4 classes: **Vol**, **Passatger**, **Reserva** i **Aerolinia**¹. També desenvoluparem un programa que provarà totes les funcionalitats previstes.

La [primera tasca](#) d'aquesta activitat serà especificar les classes **Vol**, **Passatger**, **Reserva** i **Aerolinia** com es descriuen a continuació.

Un **Vol** s'identifica amb un número de vol, té una ciutat d'origen, una ciutat destinació, l'hora de sortida prevista, l'hora d'arribada estimada, l'estat del vol (a l'hora, retardat, etc.) i el nombre de seients disponibles. Exemple de com es mostren les dades d'un **Vol**:

```
Número vol: AI101
Ciutat origen: Delhi
Ciutat destinació: London
Hora sortida: 10:00
Hora arribada: 15:00
Estat vol: On-time
Seients disponibles: 200
```

¹Farem servir la paraula **Aerolinia**, en lloc d'**Aerolínia**, per evitar l'accent en el nom de la classe.



Un **Passatger** es representa amb un nom, que l'identifica, l'estat de la seva reserva (**Confirmed**, **Pending** o **Hold**) i el número de seient assignat. Exemple de com es mostren les dades d'una instància de **Passatger**:

```
Nom passatger: Alice-Brown
Estat reserva: Confirmed
Número seient: A2
```

Una **Reserva** s'identifica amb un número de reserva, i també hi consten el nom del/de la passatger/a, el número de vol i el mètode de pagament que ha fet servir en el procés de reserva. Exemple de com es mostren les dades d'una **Reserva**:

```
Número reserva: B001
Nom passatger: John-Doe
Número vol: AI101
Pagament: Credit-Card
```

Una **Aerolinia** s'identifica amb el seu nom i disposa d'un conjunt de vols i un conjunt de reserves. Ambdós conjunts estan limitats en volum i, per això, cal indicar tant el nombre màxim de vols com el nombre màxim de reserves permesos.

Per cada classe, **haureu de definir**: almenys un constructor, un destructor, modificadors dels atributs (quan sigui necessari), consultors per accedir als atributs i operacions de lectura i escriptura.

Recordeu que, en cridar un mètode, haureu de comprovar primer si se satisfà la **Pre** i, en cas contrari, decidiu què ha de fer l'aplicació per tractar aquesta situació.

La **segona tasca** serà escriure un programa simple que permeti gestionar una sistema de gestió de reserves de vols com el que acabem de descriure.

L'aplicació haurà de llegir primer les dades de l'aerolínia però, per la PoC, només treballarem amb l'aerolínia "Voling", per tant, sols haurà de llegir el nombre màxim de vols i el nombre màxim de reserves permesos per aquesta aerolínia. Després oferirà la possibilitat de provar les operacions següents:

1. Enregistrar un vol [opció **enregistrarVol**].

A l'hora d'enregistrar un vol, cal comprovar que no hi hagi cap altre vol amb el mateix número i que no se superi el nombre màxim de vols de l'aerolínia. Si se satisfan les dues condicions, s'enregistra el vol i es mostra el missatge **OK**; altrament, el programa ha de mostrar un missatge d'error segons correspongui.



2. Enregistrar una reserva [opció `enregistrarReserva`].

Quan es fa la reserva d'un vol, cal comprovar que: *a)* no hi hagi cap altra reserva amb el mateix número; *b)* no se superi el nombre màxim de reserves de l'aerolínia; *c)* el número de vol existeixi, i *d)* el vol tingui seients disponibles. Suposarem que el seient assignat en una reserva sempre és correcte, és a dir, que no ha estat assignat prèviament i és un número de seient vàlid. També suposarem que un mateix passatger no fa més d'una reserva per un mateix vol. Si se satisfan les condicions anteriors, s'enregistra la reserva, s'actualitza el nombre de seients disponibles del vol i es mostra el missatge OK. En cas de no satisfer-se alguna de les condicions, el programa ha de mostrar un missatge d'error segons correspongui.

3. Mostrar tots els vols enregistrats al sistema [opció `mostrarVols`].

Es mostren les dades de tots els vols segons l'ordre en què s'han enregistrat al sistema. Per cada vol es mostren: el seu número de vol, la ciutat d'origen, la ciutat destinació, l'hora de sortida, l'hora d'arribada, l'estat del vol i el nombre de seients disponibles. Si no hi ha cap vol enregistrat en el sistema, el programa ha de mostrar un missatge d'error.

4. Mostrar totes les reserves enregistrades al sistema [opció `mostrarReserves`].

Es mostren les dades de totes les reserves **ordenades per número de reserva** de forma creixent. Per cada reserva es mostren: el seu número de reserva, el nom del passatger que ha fet la reserva, el número de vol i el mètode de pagament que ha fet servir. Si no hi ha cap reserva enregistrada en el sistema, el programa ha de mostrar un missatge d'error.

5. Mostrar tots els passatgers d'un vol [opció `mostrarPassatgersVol`].

Es mostren les dades de tots els passatgers que han reservat un vol amb el número de vol donat d'entrada. La llista de passatgers es mostra en **ordre alfabètic**. Si no hi ha cap vol enregistrat en el sistema amb el número de vol donat o bé no hi ha cap reserva feta per aquest número de vol, el programa ha de mostrar un missatge d'error segons correspongui.

6. Mostrar totes les reserves que es trobin en un estat donat [opció `mostrarReservesEstat`].

Es mostren les dades de totes les reserves enregistrades al sistema que es trobin en un determinat estat donat d'entrada (**Confirmed**, **Pending** o **Hold**). La llista de reserves es mostra **ordenada per número de reserva** de forma creixent. Si no hi ha cap reserva enregistrada en el sistema o bé no hi ha cap reserva que es trobi en l'estat donat d'entrada, el programa ha de mostrar un missatge d'error segons correspongui.

7. Sortir del programa [opció `acabar`].



Es garanteix que el fitxer d'entrada sempre contindrà aquesta opció.

Qüestions a tenir en compte:

- Les dades del sistema de reserves de vols només es mantenen durant l'execució del programa, és a dir, no es guarden en cap fitxer extern.
- El sistema treballa per una única aerolínia ("Voling") que compta amb dades diferents a cada execució. El nombre màxim de vols i el nombre màxim de reserves permesos es llegeixen al principi de l'execució i una única vegada. Es garanteix que els fitxers d'entrada sempre contindran aquests dos valors.
- No hi ha operacions de cancel·lacions de vols ni d'anul·lacions de reserves durant l'execució.
- Les dades d'un passatger estan sempre associades a una reserva. No hi ha operacions específiques sobre els passatgers.

Atès que encara no heu implementat les classes, aquest programa farà ús de les especificacions anteriors però, fins que no les implementeu, no podreu executar-lo. Per tant, la [tercera tasca](#) serà implementar les classes i generar el programa executable.

La [quarta tasca](#) serà provar l'executable amb els jocs de prova que us facilitem.

2 Exemples de sortida

A continuació es mostren alguns exemples de sortida de les operacions demanades. Per obtenir els detalls de totes els casos, mireu-vos els fitxers `sampleN.inp` i els seus corresponents `sampleN.cor` per comprovar el format de les entrades i les sortides esperades en **totes les situacions previstes en la prova de concepte²**.

1. Enregistrar un vol [opció `enregistrarVol`].

```
Enregistrament vol: AI101
OK
Enregistrament vol: AI101
Error: AI101 vol ja existent a Voling, no s'ha afegit el vol
...
Enregistrament vol: AF663
Error: el vol supera el nombre màxim de vols a Voling, no s'ha
afegit el vol
```

²En un sistema real segur que es podrien donar moltes altres situacions que caldria controlar.



2. Enregistrar una reserva [opció `enregistrarReserva`].

```
Enregistrament reserva: A001
OK
...
Enregistrament reserva: C001
Error: vol BA222 no existent a Voling, no s'ha afegit la reserva
```

3. Mostrar tots les reserves enregistrades al sistema [opció `mostrarReserves`]

```
Reserves de Voling (per número de reserva):
1.
Número reserva: A001
Nom passatger: Chris-Martin
Número vol: AF662
Pagament: Credit-Card
2.
Número reserva: A002
Nom passatger: Jane-Devon
Número vol: AF663
Pagament: Debit-Card
```

4. Mostrar tots els passatgers d'un vol [opció `mostrarPassatgersVol`].

```
Passatgers del vol AI101 de Voling (en ordre alfabètic):
1.
Nom passatger: Alice-Brown
Estat reserva: Confirmed
Número seient: A2
2.
Nom passatger: Jane-Devon
Estat reserva: Hold
Número seient: F2
3.
Nom passatger: John-Doe
Estat reserva: Confirmed
Número seient: A1
```



5. Mostrar totes les reserves que es trobin en un estat donat [opció `mostrarReservesEstat`].

Reserves de Voling amb estat "Pending" (per número de reserva):

1.

Número reserva: A001

Nom passatger: Chris-Martin

Número vol: AF662

Pagament: Credit-Card

6. Mostrar tots els vols enregistrats al sistema [opció `mostrarVols`].

Vols de Voling (per ordre d'enregistrament):

1.

Número vol: AI101

Ciutat origen: Delhi

Ciutat destinació: London

Hora sortida: 7:15

Hora arribada: 11:30

Estat vol: On-time

Seients disponibles: 200

2.

Número vol: AI102

Ciutat origen: London

Ciutat destinació: Delhi

Hora sortida: 8:45

Hora arribada: 22:30

Estat vol: Scheduled

Seients disponibles: 200

3.

Número vol: AF662

Ciutat origen: París

Ciutat destinació: Dubai

Hora sortida: 10:15

Hora arribada: 20:04

Estat vol: Arrived

Seients disponibles: 120



3 Comentaris pel lliurament

- El codi de cada mètode ha d'estar comentat descrivint en quines condicions és aplicable (precondició) i quin és el resultat final (postcondició).
- Per definir les classes `Vol`, `Passatger`, `Reserva` i `Aerolinia` no podeu fer servir classes diferents a les que heu vist fins ara al laboratori o a teoria.
- Disposareu de diversos jocs de prova públics. Per cada un d'ells hi haurà un arxiu d'entrada i un altre amb la sortida corresponent. Heu de respectar el format de les entrades i les sortides que apareixen en aquests arxius.
- També disposareu d'un Makefile per facilitar la compilació, muntatge i generació del programa executable. El vostre executable s'ha de poder generar fent servir aquest Makefile que us donem.
- El que haureu de lliurar és un arxiu comprimit (.zip o .rar) amb tots els arxius *.hpp i *.cpp que siguin necessaris per compilar i muntar el vostre programa (**NOMÉS** aquests).
- Dins d'Atenea trobareu una Tasca on penjar el vostre arxiu. No s'acceptaran treballs enviats fora de termini ni enviats de cap altra manera.

Data límit: diumenge 24 de març de 2024 a les 23 h 59 min
--

-
- Per valorar l'activitat es tindran en compte tant l'especificació com la implementació així com la superació de diverses execucions.
 - Per aquesta activitat, heu de formar equips de **dues persones**.
 - En el cas de detecció de còpia entre dos o més equips, la normativa de l'EPSEVG determina que la qualificació tant de l'activitat com de l'assignatura serà de SUSPENS (0) per totes les persones integrants de tots els equips.

Les accions irregulars que poden conduir a una variació significativa de la qualificació d'un o més estudiants constitueixen una realització fraudulenta d'un acte d'avaluació. Aquesta acció comporta la qualificació descriptiva de suspens i numèrica de 0 de l'acte d'avaluació i de l'assignatura, sense perjudici del procés disciplinari que es pugui derivar com a conseqüència dels actes realitzats.