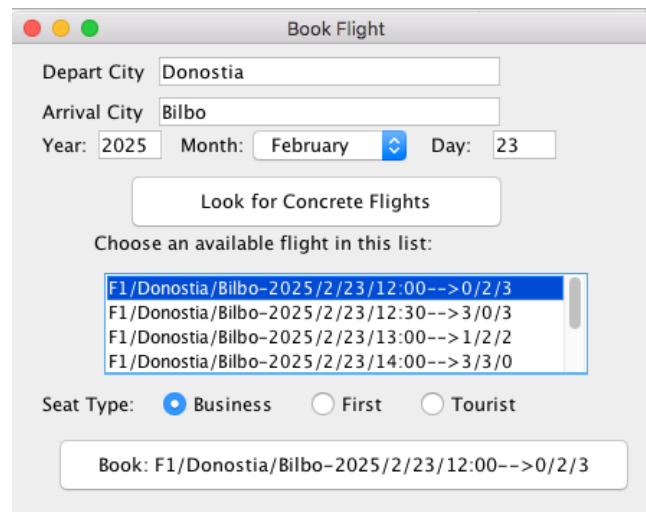


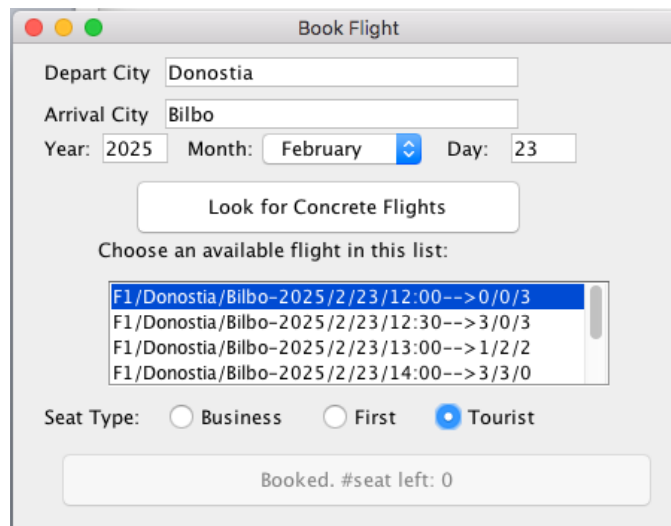
## ***ERNE: Ez dago pertsistentziarik “ESERLEKU ERRESERBATU” funtzionalitatean***

“FlightBooking” erabilpen kasuan erabiltzen diren objektuen pertsistentziaren kudeaketa Sinking Soft-ean konpondu behar duzuen 2. marroia da. Sesio batean erreserbatutako hegaldi konkretuak EZ dira gordetzen hurrengo sesioetan.

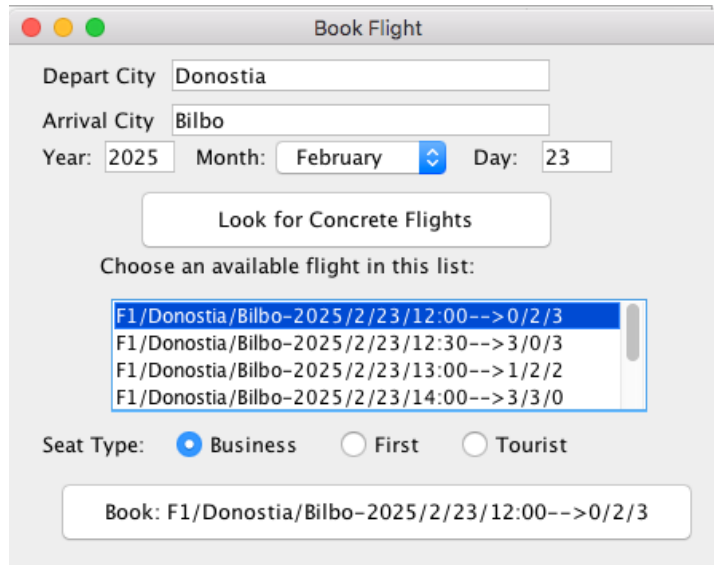
Adibidez, hegaldiak erreserbarako klasea exekutatzen baduzu, eta 2025/2/23 12:00ko hegaldian geratzen diren 2 First eserleku erreserbatzen badituzu, (Datuak sartu, “Look for Concrete Flights” sakatu, “First” aukeratu eta “Book...” sakatu bi aldiz).



“First” eserlekuak bukatu direla hegaldi horrentzat ikusi daiteke.



Baina, aplikazioa berriz itxi eta irekitzen badugu (aurreko JFrame-a itxiz), hegaldi konkretu horren eserlekuak berriz erreserbatzea posible dela ikusi dezakegu, hurrengo pantailan ikusten den bezala.



Gertatu dena, ConcreteFlight klasearen objektua memoria nagusian eguneratzen dela aplikazioa exekutatzen den baloitzean, eta aplikazio berriz exekutatzerakoan, hegaldi guztien balioak berriz hasieratu egiten dira.

```
f1.addConcreteFlight("CF2",newDate(2025,1,23),3,0,3,"12:00");
```

Gure aplikazioak aurreko balioak "gogoratu" nahiko balitu, objektuak memoria finko batean gorde egin beharko lituzke (hau da, datu base batean) lehendabiziko exekuzioan. Horrela, bigarrenko exekuzioan posible izango litzateke objektuak datubasetik berreskuratzea, eta modu horretan, eguneratutako hegaldiekin lan egin.

#### Hurrengo eskatzen da:

- 1) Erreserba bat egin ondoren, datu basea eguneratzen da hegaldi konkretu horretan geratzen diren eserleku libreekin. Hau egiteko, aplikazioa exekutatu baino lehen datu base bat sortu beharko duzu hegaldi konkretu batzuekin.

#### Taldeko Lana.

Marroia egiteko, hasierako kodetik garatu daiteke ([github](#) kodea), edo lehendabiziko marroiaren ebazpenetik.

Entregatzeko data: **Otsailak 11, asteartea (24:00). Ez da luzapenik egongo**



## Entregatu behar dena.

Zip fitxategi bat kodearekin eta erroan **readme.txt** fitxategi bat hurrengo informazioarekin:

- Lortu duzu arazoa konpontzea (BAI/EZ/PARTZIALKI)?  
PARTZIALKI/EZ bada komentatu izan duzun arazoa.
- Zenbat ordu behar izan dituzue talde osoa marroia konpontzeko?
- Gehienez 3'ko bideo baten esteka kodean egin dituzun aldaketak azalduz, eta aplikazioren demo batekin.

## GARAPEN OHARRAK:

Objektuak datubasetik modu egoki batean gorde eta berreskuratuzeko kontuan hartu hurrengo **hiru ohar**:

1. Flight eta ConcreteFlight guztiak identifikadore (gako) bat eduki behar dute "String" motakoa. Adibidez:

**Flight klasean**

@Id

public String flightCode;

**ConcreteFlight klasean**

@Id

public String concreteFlightCode;

2. Flight klasean, hurrengo anotazioa ipini "concreteFlights" atributuaren gainean:

```
@OneToMany(fetch=FetchType.EAGER, cascade = CascadeType.PERSIST)  
Collection<ConcreteFlight> concreteFlights;
```

FetchType.EAGER esan nahi du, Flight bat berreskuratzen duzunean DB-tik, bere ConcreteFlight objektuak ere berreskuratuko direla.

CascadeType.PERSIST esan nahi du, Flight bat gordetzen duzunean (persist), bere "concreteFlight" objektuak ere datubasean gordeko direla.

3. Modu berean ConcreteFlight klasean, hurrengo anotazio "flight" atributuari ipini

```
@OneToOne(fetch=FetchType.EAGER, cascade = CascadeType.PERSIST)  
Flight flight;
```