

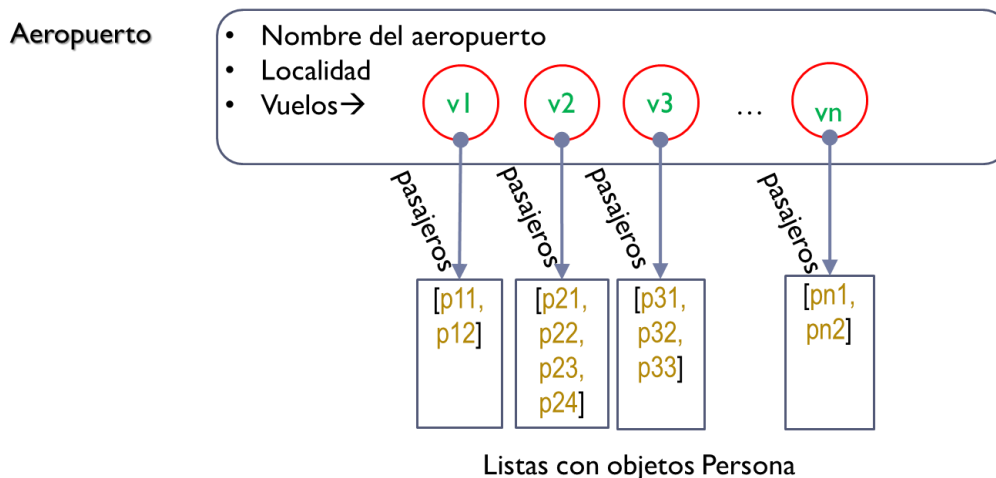


EL TIPO PERSONA, VUELO y AEROPUERTO

Consideraciones Iniciales

- Este es un enunciado que se irá incrementando a medida que avancemos en la teoría del 2º cuatrimestre del curso.
- Es **MUY IMPORTANTE** llevarlo al día, porque **numerosas cuestiones** avanzarán a partir de lo realizado en las anteriores, bien porque se ha resuelto en clase o se han dejado como tarea para casa.

Esquema gráfico de la estructura de datos del proyecto



EJERCICIOS:

1. Importe el proyecto: T03_Aeropuerto. (vea las instrucciones que se facilitan en la sesión de hoy)
2. En fp.aeropuerto implemente un tipo **Persona** con las siguientes características:

PERSONA

Propiedades:

- String **dni**. Consultable.
- String **nombre**. Consultable.
- String **apellidos**. Consultable.
- LocalDate fechaNacimiento. Consultable

Métodos Constructores:

- Un constructor a partir de cada uno de los atributos (constructor canónico).
- Un constructor con apellidos y nombre en el que el dni quede con la cadena vacía ("") y la fecha de nacimiento el 1 de enero de 1995.

Restricciones:

- Ningún atributo puede ser nulo.
- La fecha de nacimiento no puede ser mayor que la actual

El criterio de igualdad:

- Dos personas son iguales si tienen el mismo dni.

El criterio de ordenación:



- Por dni.

La representación como cadena:

- Todos los atributos básicos con el nombre del tipo y separados por comas.

3. En `fp.aeropuerto.test` ya tiene implementado una clase **TestPersona01** con algunos datos por cumplimentar. Si lo hace bien, al ejecutar el test debe salir lo siguiente:

```
Test constructor-1 con todos los parámetros
    Persona[dni=12345678A, nombre=Manolito, apellidos=Gafotas, fechaNacimiento=2000-01-15]
    Construido correctamente!!

Test constructor-1 con todos los parámetros nulos
    Ha habido un error al construir la persona
    java.lang.IllegalArgumentException: fp.aeropuerto.Persona.<init>: el parámetro 1
    es nulo

Test constructor-2 con apellidos y nombre
    Persona[dni=, nombre=Gafotas, apellidos=Manolito, fechaNacimiento=1995-01-01]
    Construido correctamente!!

Test constructor-2 con parámetros nulos
    Ha habido un error al construir la persona
    java.lang.IllegalArgumentException: fp.aeropuerto.Persona.<init>: el parámetro 2
    es nulo
```

4. Implemente en `fp.aeropuerto` el Tipo Vuelo con los siguientes requisitos:

VUELO

Propiedades:

- String **código**. Consultable. Indica el código del vuelo.
- String **destino**. Consultable. Ciudad de destino del vuelo
- LocalDateTime **fechaHoraSalida**. Consultable. Fecha y hora de salida del vuelo.
- Duration **duración**. Consultable.
- Double **velocidad**. Consultable.
- Double **precio**. Consultable. Precio del billete.
- Integer **númeroPlazas**. Consultable. Número de plazas del vuelo.
- Boolean **conEscalas**. Consultable. Indica si el vuelo tiene, o no, escalas.
- List<Persona> **pasajeros**. Consultable. Lista con los pasajeros del vuelo.
- Integer **númeroPasajeros**. Consultable.
- Boolean **vueloCompleto**. Consultable. Indica si el vuelo está, o no, completo.
- Double **porcentajeOcupación**. Consultable.
- LocalDateTime **fechaHoraLlegada**. Consultable. Fecha y hora de llegada a destino.
- Compañía **compañía**. Se calcula a partir de los 3 primeros caracteres del código. Podrá tomar uno de los cuatro siguientes valores: IBE, LUF, RYA, VGL.

Métodos Constructores:

- Un constructor a partir de cada uno de los atributos (constructor canónico).



Restricciones:

- Ningún atributo puede ser nulo.
- El código debe tener 7 caracteres, los 3 primeros alfabéticos y los 3 último numéricos. El del centro serán un guion (-). Si hay error el mensaje es: "El formato del código debe ser aaa-nnn"
- El precio es mayor o igual que cero.
- El número de plazas debe ser mayor o igual que cero.
- La velocidad debe ser mayor o igual que cero.
- La duración es mayor que cero.
- El número de pasajeros es menor o igual que el número de plazas.

El criterio de igualdad

- Dos vuelos son iguales si tienen el mismo código, destino y la misma fecha de salida (sin la hora)

El criterio de ordenación:

- Por el código y desempatan por el destino y, en su caso, también por la fecha de salida (sin la hora)

La representación como cadena:

- El código, el destino, fecha y hora de salida, fecha y hora de llegada, número de plazas y número de pasajeros.

5. En `fp.aeropuerto.test` ya tiene implementado una clase **TestVuelo01** que si la ejecuta debe obtener lo siguiente:

```
Test constructor-1 con todos los parámetros
    Vuelo [código=IBE-001, destino=Valencia, fechaHoraSalida=2023-08-01T09:00,
    fechaHoraLlegada=2023-08-01T10:55 , Nro.pasajeros=3]
    Construido correctamente!!

Test constructor-1 con algún parámetro nulo
    Ha habido un error al construir el vuelo
    java.lang.NullPointerException: Cannot invoke
    "java.util.Collection.isEmpty()" because "coll" is null

Test constructor-1 con el código malo
    Ha habido un error al construir el vuelo
    java.lang.IllegalArgumentException: fp.aeropuerto.Vuelo.<init>: El formato
    del código debe ser aaa-nnn

Test constructor-1 con la duración negativa
    Ha habido un error al construir el vuelo
    java.lang.IllegalArgumentException: fp.aeropuerto.Vuelo.<init>: La duración
    debe ser >0

Test constructor-1 con la velocidad negativa
    Ha habido un error al construir el vuelo
    java.lang.IllegalArgumentException: fp.aeropuerto.Vuelo.<init>: La
    velocidad debe ser >=0

Test constructor-1 con el precio negativo
    Ha habido un error al construir el vuelo
    java.lang.IllegalArgumentException: fp.aeropuerto.Vuelo.<init>: El precio
    del billete debe ser >=0

Test constructor-1 con las plazas negativas
    Ha habido un error al construir el vuelo
```



[java.lang.IllegalArgumentException](#): fp.aeropuerto.Vuelo.<init>: El número de plazas debe ser >0

Test constructor-1 con las plazas menores que el número de pasajeros
Ha habido un error al construir el vuelo
[java.lang.IllegalArgumentException](#): fp.aeropuerto.Vuelo.<init>: El número de pasajeros debe ser <= número de plazas

Test constructor-2 con todos los parámetros
Vuelo [código=IBE-001, destino=Valencia, fechaHoraSalida=2023-08-01T09:00, fechaHoraLlegada=2023-08-01T10:55 , Nro.pasajeros=0]
Construido correctamente!!

Test constructor-2 con algún parámetro nulo
Ha habido un error al construir el vuelo
[java.lang.IllegalArgumentException](#): fp.aeropuerto.Vuelo.<init>: el parámetro 1 es nulo

Test constructor-2 con el código malo
Ha habido un error al construir el vuelo
[java.lang.IllegalArgumentException](#): fp.aeropuerto.Vuelo.<init>: El formato del código debe ser aaa-nnn

Test constructor-2 con la duración negativa
Ha habido un error al construir el vuelo
[java.lang.IllegalArgumentException](#): fp.aeropuerto.Vuelo.<init>: La duración debe ser >0

Test constructor-2 con la velocidad negativa
Ha habido un error al construir el vuelo
[java.lang.IllegalArgumentException](#): fp.aeropuerto.Vuelo.<init>: La velocidad debe ser >=0

Test constructor-2 con el precio negativo
Ha habido un error al construir el vuelo
[java.lang.IllegalArgumentException](#): fp.aeropuerto.Vuelo.<init>: El precio del billete debe ser >=0

Test constructor-2 con las plazas negativas
Ha habido un error al construir el vuelo
[java.lang.IllegalArgumentException](#): fp.aeropuerto.Vuelo.<init>: El número de plazas debe ser >0