

Problema Aeropuerto: Diseño de Tipos

Miguel Toro

Se dispone de los datos de aeropuertos en el fichero Aeropuertos.csv. Cada línea contiene , *nombre, pais, codigo, ciudad*. Las líneas son de la forma

Tirana Airport,Albania,TIA,Tirana
Berlín Brandeburgo Airport,Alemania,BER,Berlin
Bremen Airport,Alemania,BRE,Bremen
Colonia Bonn Airport,Alemania,CGN,Colonia

De aerolíneas dispones del fichero Aerolíneas.csv. Cada línea contiene podemos obtener el *codigo, nombre* en la forma

AA,American Airlines
CO,Continental Airlines,Inc.
DL,Delta Airlines Inc.
N7,National Airlines Inc.

El fichero Vuelos.csv contiene en cada línea el *codigoAerolinea, numero, codigoDestino, codigoOrigen, precio, numPlazas, duracion, hora, diaSemana*. Las líneas son de la forma

TP,0705,BER,KTW,294,21,170,287,14:50,FRIDAY
FS,0596,TZX,AAR,761,64,49,45,07:54,THURSDAY
FJ,0612,BHD,TPS,113,98,128,180,16:41,MONDAY
CX,0930,LJU,NCL,741,17,11,159,02:40,WEDNESDAY

El fichero OcupacionesVuelos.csv contiene en cada línea *codigoVuelo, fecha, InumPasajeros* de la forma

NH0818,2020-04-13 16:43:00,7
PE0174,2020-11-17 09:03:00,89
5Z0373,2020-05-16 01:46:00,64
7F0434,2020-10-01 03:24:00,94

Queremos diseñar un programa que pueda llevar a cabo entre otros posibles los *cálculos* sobre los vuelos como los siguientes:

1. Si existe un vuelo en una fecha dada Dado a un conjunto de destinos dado
2. Encontrar los destinos en una fecha dada
3. Encontrar el número total de pasajeros de los destinos que tienen un prefijo dado
4. Encontrar una relación ordenada de destinos con su número de pasajeros de llegada en un año dado.
5. Dado un destino encontrar el código del primer vuelo con plazas libres a ese destino.
6. Encontrar el destino con mayor número de vuelos de entrada y salida

7. Encontrar una relación que asocie a cada fecha la lista de los destinos de los n vuelos de la mayor duración
8. Obtener un conjunto ordenado con las duraciones de todos los vuelos cuya duración es mayor que un número de minutos dado
9. Dada una fecha f encontrar el precio medio de los vuelos con salida posterior a f. Si no hubiera vuelos devuelve 0.0
10. Obtener un conjunto con los n destinos de los vuelos con mayor duración.
11. Obtener una relación de los destinos junto a la media de los precios de los vuelos a ese destino.
12. Obtener una relación de destinos junto con las fechas de los vuelos a ese destino.
13. Obtener los n destinos que con más vuelos
14. Obtener los destinos que tienen más de n vuelos
15. Obtener una relación de destinos junto con el porcentaje de vuelos que van a ese destino.
16. Obtener una relación de destinos junto con el vuelo más barato a ese destino.
17. Obtener una relación de destinos junto a las fechas posibles a ese destino.
18.

Los tipos que vamos a necesitar son:

Tipos:

Aeropuerto, Inmutable

Propiedades:

- Código: String, básica, se compone de tres caracteres, Identifica el aeropuerto de manera única
- Ciudad: String, básica
- País: String, básica
- Nombre: String, básica

Representación:

Código, Ciudad, País, Nombre

Igualdad: Si tienen el mismo código. Aunque con la restricción impuesta al código todas las propiedades básicas serán iguales.

Métodos de Factoría:

- parse(text:String): Aeropuerto
- of(Código: String,Ciudad: String,País: String,Nombre: String): Aeropuerto

Aeropuertos

Propiedades:

- aeropuertos: List<Aeropuerto>, básica, los códigos de los aeropuertos deben ser diferentes
- codigosAeropuertos: Map<String,Aeropuerto>, derivada, código-aeropuerto
- ciudadDeAeropuerto: Map<String,String>, derivada, código-ciudad
- aeropuertosEnCiudad: Map<String,Set<Aeropuerto>>, derivada, ciudad-{codigosAeropuerto, ...}
- numAeropuertos: Integer

Operaciones:

- addAeropuerto(v: Aeropuerto):void, Añade un aeropuerto
- removeAeropuerto(v: Aeropuerto):void, Elimina un aeropuerto

Representación:

Lista de aeropuertos uno por línea

Métodos de Factoría:

- leeAeropuertos(fichero:String), void, lee de un fichero las propiedades básicas

De este tipo construiremos un solo objeto que llamaremos *datos*.

Aerolinea, Inmutable

Propiedades:

- Código: String, básica, se compone de dos caracteres, Identifica el aeropuerto de manera única
- Nombre: String

Representación:

- Codigo,Nombre

Igualdad: Si tienen el mismo código. Aunque con la restricción impuesta al código todas las propiedades básicas serán iguales

Métodos de Factoría:

- parse(text:String): Aerolinea
- of(Codigo: String,Nombre: String): Aerolinea

Aerolineas

Propiedades:

- aerolineas: List<Aerolinea>, básica, los códigos de las aerolíneas deben ser diferentes
- codigosAerolineas: Map<String,Aerolinea>, derivada, código-aerolinea
- numAeropuertos: Integer, derivada

Operaciones:

- addAerolinea(v: Aerolinea):void, Añade una aerolinea
- removeAerolinea(v: Aerolinea):void, Elimina una aerolinea

Representación:

Lista de aerolíneas una por línea

Métodos de Factoría:

- leeAerolineas(fichero:String), void, lee de un fichero las propiedades básicas

De este tipo construiremos un solo objeto que llamaremos *datos*.

Vuelo, Inmutable

Propiedades:

- CódigoAerolinea: String, básica
- Numero: String, básica, cuatro caracteres que representan un número
- CódigoDestino: String, básica, código del aeropuerto destino
- CódigoOrigen: String, básica, código del aeropuerto origen
- Precio: Integer, básica, debe ser mayor que cero
- NumPlazas: Integer, básica, debe ser mayor que cero
- Duración: Duration, básica, duración del vuelo
- Hora: LocalTime, básica,
- DíaSemana, DayOfWeek, básica
- CiudadDestino, String, derivada
- CiudadOrigen, String, derivada
- Código, String, derivada, identifica de manera única un vuelo y se compone de la concatenación del CódigoAerolinea y el Numero.

Representación:

- Código,Nombre

Igualdad: Si tienen el mismo código. Aunque con la restricción impuesta al código todas las propiedades básicas serán iguales

Métodos de Factoría:

- parse(text:String): Vuelo
- of(Código: String,Nombre: String): Aerolínea
- random(): Vuelo, un vuelo construido aleatoriamente con los aeropuertos y las aerolíneas disponibles.

Vuelos

Propiedades:

- vuelos: List<Vuelo>, básica, los códigos de los vuelos deben ser diferentes
- códigosVuelos: Map<String,Vuelo>, derivada, código-vuelo
- numVuelos: Integer, derivada

Incluir aquí como propiedades algunos de los cálculos enunciados arriba

Operaciones:

- addAerolinea(v: Aerolinea):void, Añade una aerolinea
- removeAerolinea(v: Aerolinea):void, Elimina una aerolinea

Representación:

- Lista de aerolíneas una por línea

Métodos de Factoría:

- leeAerolineas(fichero:String), void, lee de un fichero las propiedades básicas
- random(num: Integer): void, construye numVuelos

De este tipo construiremos un solo objeto que llamaremos *datos*.

OcupacionVuelo, Inmutable

Propiedades:

- CódigoVuelo: String, básica
- Fecha: LocalDateTime, básica, fecha-hora de salida
- NumPasajeros: Integer, básica, debe ser mayor o igual a cero
- Vuelo: Vuelo, derivada
- Llegada: LocalDateTime, fecha-hora de llegada
- FechaSalida: LocalDate, fecha de salida
- HoraSalida: LocalTime, hora de salida

Representación:

- CodigoVuelo, FechaSalida, HoraSalida

Igualdad: Si tienen el mismo código. Aunque con la restricción impuesta al código todas las propiedades básicas serán iguales

Métodos de Factoría:

- parse(text:String): Vuelo
- of(Codigo: String,Nombre: String): Aerolínea
- random(): Vuelo, un vuelo construido aleatoriamente con los aeropuertos y las aerolíneas disponibles.

OcupacionesVuelos

Propiedades:

- Ocupaciones: List<OcupacionVuelo>, básica
- NumeroDeOcupaciones: Integer, Derivada

Incluir aquí como propiedades algunos de los cálculos enunciados arriba

Operaciones:

- addOcupacionVuelo(oc: OcupacionVuelo):void, Añade una ocupación de un vuelo
- removeOcupacionVuelo(oc: OcupacionVuelo):void, Elimina una ocupación de un vuelo

Representación:

- Lista de ocupaciones una por línea

Métodos de Factoría:

- `leeOcupaciones(fichero:String)`, void, lee de un fichero las propiedades básicas
- `random(num: Integer, año:Integer)`: void, construye num ocupaciones en el año dado a partir de los vuelos disponibles

De este tipo construiremos un solo objeto que llamaremos *datos*.