Francisco de Asís Bermúdez Campuzano

Durante este proyecto vamos a trabajar con información (ficticia) sobre diversos conciertos realizados por distintos grupos de Rock. Para ello disponemos de un listado de datos recopilado de distintas giras.

Para cada concierto se posee el grupo autor, la ciudad donde se realizó el mismo, el tipo de guitarra principal utilizada, el álbum cuyas canciones fueron tocadas, el número total de espectadores, el coste de la entrada, el tipo de Rock, la fecha cuando se realizó el concierto además de su hora de inicio y finalmente si fue un concierto benéfico.

Los datos se encuentran almacenados en ficheros en formatos CSV codificados en UTF-8. Cada registro de un fichero ocupa una línea y contiene los datos correspondientes a un concierto. Por ejemplo, estas son las primeras líneas de un fichero con conciertos.

```
nombreGrupo,ciudad,tipoGuitarra,album,espectadores,precioEntrada,tipoRock,fecha,hora,esBenefico
AC/DC,Buffalo,GIBSON_SG,Black Ice,45000,35.00,HARD-ROCK,11/09/2016,20:00,NO
AC/DC,New York,GIBSON_SG,High Voltage,28000,20.00,HARD-ROCK,14/09/2016,18:00,SI
AC/DC,Düsseldorf,GIBSON_SG,Black Ice,20000,50.00,HARD-ROCK,15/06/2015,23:30,NO
AC/DC,Barcelona,GIBSON_SG,TNT,18000,20.00,HARD-ROCK,31/03/2009,17:00,SI
AC/DC,Londres,GIBSON_SG,High Voltage,20000,45.00,HARD-ROCK,14/04/2009,21:30,NO
```

Los tipos que forma el modelo de datos son los siguientes:

- **tipoGuitarra**: tipo enumerado con los valores posibles de las guitarras.
- **tipoRock**: tipo enumerado que determina el subgénero del rock.
- **Concierto**: tipo que representa un concierto ROCK.
- **Conciertos**: tipo contenedor que representa una colección de conciertos.

Tipo Concierto

Propiedades:

- **nombreGrupo**, de tipo String. Consultable.
- **ciudad**, de tipo String. Consultable.
- **tipoGuitarra**, de tipo enumerado. Consultable.
- **album**, de tipo String. Consultable.
- **espectadores**, de tipo Integer. Consultable.
- **precioEntrada**, de tipo Double. Consultable y modificable.
- **tipoRock**, de tipo enumerado. Consultable.
- **fecha**, de tipo LocalDate. Consultable y modificable.
- **hora**, de tipo LocalTime. Consultable y modificable.
- **esBenefico**, de tipo Boolean. Consultable.

Los atributos enumerados se componen de:

- **tipoGuitarra**: *GIBSON_SG, FENDER_TELECASTER, IBANEZ_RG, PRS_SE, EPIPHONE_LES_PAUL*
- **tipoRock**: *HARD_ROCK, PUNK_ROCK, ROCK_N_ROLL, ROCK_ANDALUZ, POP_ROCK*

Constructores:

- C1: recibe los siguientes parámetros: el nombre del grupo, la ciudad anfitriona, el tipo de guitarra, el álbum elegido, el número de espectadores, el precio de la entrada, el tipo de Rock del grupo, la fecha del concierto, la hora de inicio, especificará si es benéfico.
- C2: Un constructor donde reciba los atributos definidos a excepción del precio de la entrada, en caso de ser benéfico esta se marcará automáticamente a 25.00€.

Restricciones:

- R1: El precio de entrada en los conciertos benéficos debe ser superior a 5.00€.
- R2: Ningún concierto podrá estar fechado más tarde que el 31/12/2022.
- R3: Todo concierto deberá comenzar posterior a las 14:00 PM.

Criterio de igualdad: dos conciertos son iguales si pertenecen al mismo grupo, la misma fecha y hora planificada.

Criterio de orden natural: por fecha, a igualdad de esta por hora planificada, y a igualdad de esta por grupo.

Representación como cadena: incluirá el nombre del grupo, la fecha, la hora, el álbum y el precio de entrada.

Tipo Conciertos

Propiedades:

- Conciertos, de tipo Set<Concierto>. Consultable. Conjunto de conciertos.

Constructores:

- Un constructor sin parámetros.
- Un constructor a partir de un Stream<Concierto>.

Criterio de igualdad: dos conciertos son iguales si sus conjuntos de conciertos son iguales.

Representación como cadena: generada automáticamente con todas las propiedades básicas del tipo.

Otras operaciones:

- *void añadirConcierto(Concierto c):* añade un concierto al conjunto.
- *Integer getNumeroConciertos():* devuelve el total de conciertos registrados.

Tratamientos secuenciales:

- *Long numeroConciertoRock(TipoRock rock):* devuelve el número total de conciertos pertenecientes a un subgénero rock.
- *Boolean existenConciertoCiudad(String ciudad):* comprueba si se han realizado o están planificados uno o varios conciertos en una determinada ciudad.
- *Integer numeroEspectadoresCiudad(String ciudad):* devuelve el número total de espectadores que ha habido en una ciudad.
- *OptionalDouble mediaPrecioEntradaPorFecha(LocalDate fecha):* obtenemos la media del valor de las entradas en una determinada fecha.
- *List<Concierto> listaConciertoPorGuitarra (TipoGuitarra guitarra, Integer n):* muestra una lista de longitud n con los conciertos realizados por un determinado modelo de guitarra.
- *TipoRock calcularRockPredominanteDeCiudad(String c):* calcula el rock que más se repite en una ciudad recibida como parámetro.
- *Map<String,Double> calcularPromedioEspectadoresPorCiudadDeUnTipoRock(TipoRock r) :* realiza un diccionario cuyas claves son las ciudades y el valor es el promedio de los espectadores de un tipo de rock (recibido como parámetro), en la ciudad correspondiente.
- *List<String> listaNCiudadesMenorNumeroPromedioEspectadores(Integer n):* obtenemos una lista con las n ciudades con menor media de espectadores.
- *Concierto conciertoMasBaratoPeriodoTiempo(LocalDate f1, LocalDate f2):* devuelve el concierto mas barato dentro de un determinado periodo de tiempo compuesto por dos fechas
- *void modificarValorConciertosBeneficosGrupo(Double p, String n):* modifica el precio de la entrada de los conciertos benéficos de un determinado grupo.

Tipo FactoriaConciertos

Operaciones:

- *Conciertos leerConciertos(String fichero)*: lee un fichero de conciertos y construye un objeto de tipo Conciertos.
- *Conciertos parsearConcierto(String linea)*: crea un objeto de tipo Concierto a partir de una cadena de caracteres. La cadena de caracteres debe tener el mismo formato que las líneas del fichero CSV.