Ejercicio: Servicio de alquiler de bicicletas públicas de Sevilla (Sevici)

Autor: Mariano González. **Revisores**: José A. Troyano, Fermín Cruz. **Última modificación**: 5/3/2020

En este ejercicio vamos a trabajar con datos de la red de estaciones del servicio de alquiler de bicicletas de Sevilla (Sevici), disponibles en https://citybik.es/. El objetivo del ejercicio es leer estos datos y realizar distintas operaciones con ellos.

Los datos se encuentran almacenados en un fichero en formato CSV codificado en UTF-8. Cada registro del fichero ocupa una línea y contiene los datos correspondientes a una estación: nombre de la estación, número total de puestos de la estación, número de bicicletas disponibles y latitud y longitud donde se ubica la estación.

Estas son las primeras líneas del fichero. La primera línea es una cabecera que contiene los nombres de los campos del registro:

```
name,slots,free_bikes,latitude,longitude
149_CALLE ARROYO,20,11,37.397829929383,-5.97567172039552
257_TORRES ALBARRACIN,20,4,37.38376948792722,-5.908921914235877
243_GLORIETA DEL PRIMERO DE MAYO,15,9,37.380439481169994,-5.953481197462845
109_AVENIDA SAN FRANCISCO JAVIER,15,14,37.37988413609134,-5.974382770011586
073_PLAZA SAN AGUSTIN,15,4,37.38951386231434,-5.984362789545622
```

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

- **Hemisferio**: tipo enumerado con los dos valores posibles de un hemisferio terrestre.
- **Coordenadas**: ubicación geográfica de una estación.
- **Estación**: tipo que representa una estación de la red.
- **RedEstaciones**: tipo contenedor que representa una red de estaciones de alquiler de bicicletas.
- **FactoriaRedEstaciones**: tipo factoría para cargar la red de estaciones del fichero.

Tipo Coordenadas

Propiedades:

- **latitud**, de tipo Double. Consultable y modificable. Latitud de las coordenadas.
- **longitud**, de tipo Double. Consultable y modificable. Longitud de las coordenadas.
- **hemisferio**, del tipo enumerado Hemisferio, que puede tomar los valores NORTE y SUR. Consultable. Hemisferio al que pertenecen las coordenadas.

Constructores:

- Un constructor que recibe un parámetro por cada propiedad básica del tipo, en el mismo orden en el que están definidas.
- Un constructor sin parámetros que crea un objeto con latitud 0º y longitud 0º.
- Un constructor a partir de String. Ejemplo de formato de la cadena: “(-1.5, 0.22)”

Criterio de igualdad: dos coordenadas son iguales si su latitud y su longitud son iguales.

Criterio de ordenación: por latitud, y a igualdad de esta por longitud.

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

Restricciones:

- La latitud debe estar comprendida entre -90º y +90º.
- La longitud debe estar comprendida entre -180º y +180º.

Otras operaciones:

- *Double getDistancia(Coordenadas c)*. Calcula la distancia entre dos coordenadas.

Tipo Estación

Propiedades:

- **nombre**, de tipo String. Consultable. Nombre de la estación.
- **puestos**, de tipo Integer. Consultable. Número de puestos de los que dispone la estación.
- **bicis disponibles**, de tipo Integer. Consultable y modificable. Número de bicicletas disponibles para alquiler en la estación.
- **ubicación**, de tipo Coordenadas. Consultable. Coordenadas geográficas de la estación.
- **puestos vacíos**, de tipo Integer. Consultable. Número de puestos vacíos de la estación, que se calcula como la diferencia entre el número de puestos y el número de bicicletas disponibles.
- **tiene bicis disponibles**, de tipo Boolean. Consultable. Toma valor true si la estación tiene al menos una bicicleta disponible, y false si no hay ninguna.

Constructores:

- Un constructor que recibe un parámetro por cada propiedad básica del tipo, en el mismo orden en el que están definidas.
- Un constructor a partir de String. Ejemplo de formato de la cadena: “087_PLAZA NUEVA,40,35,37.38896647697466,-5.995294220038549”

Criterio de igualdad: dos estaciones son iguales si sus nombres son iguales.

Criterio de ordenación: por su nombre.

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

Restricciones:

- El número de puestos debe ser mayor que 0.
- El número de bicicletas disponibles debe ser mayor o igual que 0 y menor o igual que el número de puestos.

Tipo RedEstaciones

Propiedades:

- **estaciones**, de tipo List<Estacion>. Consultable. Lista de estaciones de la red.

Constructores:

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

Criterio de igualdad: dos redes de estaciones son iguales si sus listas de estaciones son iguales.

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

Otras operaciones:

- *void añadirEstacion(Estacion e)*: añade una estación a la red.
- *List<Estacion> getEstaciones()*: obtiene la lista de estaciones de la red.
- *Integer getNumeroEstaciones()*: obtiene el número de estaciones de la red.

Tratamientos secuenciales:

- *List<Estacion> getEstacionesBicisDisponibles()*: obtiene una lista con las estaciones que tienen alguna bicicleta disponible.
- *List<Estacion> getEstacionesBicisDisponibles(int k)*: obtiene una lista con las estaciones que tienen un número mínimo de bicicletas disponibles.
- *SortedSet<Estacion> getEstacionesCercanas(Coordenadas cs, double d)*: obtiene un conjunto ordenado con las estaciones que tienen alguna bicicleta disponible y que están cerca de una ubicación dada.
- *Set<Coordenadas> getUbicacionEstaciones()*: obtiene un conjunto con la ubicación de todas las estaciones.
- *Set<Coordenadas> getUbicacionEstacionesDisponibles(int k)*: obtiene un conjunto con la ubicación de las estaciones que tienen un número mínimo de bicicletas disponibles.
- *Estacion getEstacionMasBicisDisponibles()*: obtiene la estación que tiene más bicicletas disponibles.

Tipo FactoriaRedEstaciones

Operaciones:

- *RedEstaciones leerRedEstaciones(String nombreFichero)*: lee un fichero de estaciones y construye un objeto de tipo RedEstaciones.
- *Estacion parsearEstacion(String lineaCSV)*: crea un objeto de tipo Estación a partir de una cadena de caracteres. La cadena de caracteres debe tener el mismo formato que las líneas del fichero CSV.