

Ejercicio. Ciclismo

Se quiere implementar una aplicación para realizar estadísticas sobre los ganadores de carreras ciclistas como el Tour de Francia, el Giro de Italia o la Vuelta Ciclista a España. Para ello se diseñan:

- Un tipo base llamado **Ganador** que representa el ganador de una edición de una carrera ciclista;
- Un tipo contenedor, llamado **EstadisticasCarrera**, que agrupa a los ganadores de las distintas ediciones de la carrera y permite obtener estadísticas sobre la misma.
- Una factoría **FactoriaCiclismo** para crear los objetos de este dominio.

Suponga definido el tipo **Ganador** con las propiedades y métodos definidos en la interfaz **Ganador**:

```
public interface Ganador {  
    Integer getAnyo(); //Año en que gana la carrera  
    String getNacionalidad(); //Nacionalidad del ganador  
    String getNombre(); //Nombre del ganador  
    String getEquipo(); //Equipo con el que gana la carrera  
    Integer getKmRecorridos(); //Kms. Totales recorridos en la carrera  
    Duration getTiempoEmpleado(); //Tiempo empleado para hacer el recorrido  
    Integer getNumEtapasGanadas(); //Número de etapas ganadas en la carrera  
    Integer getNumDiasMaillot(); //Número de días con el maillot de ganador  
    Double getVelocidadMedia(); //Velocidad media en la carrera  
}
```

Implemente en el tipo **EstadisticasCarrera**, que tiene la siguiente descripción, los métodos definidos como otras operaciones¹:

Propiedades:

- **Nombre carrera:** String. Básica, consultable. Representa el nombre de la carrera.
- **ganadores:** List<Ganador>. Básica, consultable. Representa la lista de ganadores de la carrera.

Constructores:

- **C1:** Constructor con un parámetro de tipo String para el nombre de la carrera, con el que se crea una carrera sin ganadores.
- **C2:** Constructor con dos parámetros: el nombre de la carrera y una colección de ganadores de la misma.

Representación como cadena:

- El nombre de la carrera, seguido de un guion, y la representación como cadena de los ganadores entre corchetes y separados por comas.

Criterio de igualdad:

- Dos objetos de tipo EstadisticaCarrera son iguales si tienen el mismo nombre y los mismos ganadores.

Otras operaciones:

- **List<String> getGanadoresConRecorridoInferiorA(Integer km):** Devuelve una lista con los nombres de los ganadores que han recorrido menos kilómetros que los dados como parámetro.
- **Boolean hanGanadoTodosAlgunaEtapa():** Devuelve *true* si todos los ganadores han ganado alguna etapa en la edición en la que ganaron la carrera.
- **Long getNumeroGanadores():** Devuelve el número de ganadores distintos de la carrera. Si un ciclista ha ganado el tour en varias ediciones, solo se debe contar una vez.
- **Set<String> getEquiposGanadores():** Devuelve un conjunto con los nombres de los equipos que han ganado la carrera.
- **Ganador buscaGanador(String nombre):** Devuelve el primer ganador que concuerde con el nombre dado como parámetro. Si no se encuentra ninguno, se devuelve null.
- **Ganador buscaGanador(Integer anyo):** Devuelve el ciclista ganador de ese año, si es que existe. Si no se encuentra ninguno, se devuelve null.

¹ Ayúdese de los comentarios //TODO definidos en el proyecto asociado al ejercicio

- ***Integer calculaDistanciaTotal()***: Devuelve la suma de los kilómetros recorridos en todas las ediciones de la carrera.
- ***Integer getKmMenorRecorrido()***: Devuelve el número de kilómetros de la edición en la que se ha hecho el recorrido más corto. Si no se puede calcular, devuelve cero.
- ***Double getMediaEtapasGanadas(String equipo)***: Devuelve el número medio de etapas que han ganado los ciclistas que corren en ese equipo. Si no se puede calcular, devuelve cero.
- ***String getGanadorMasRapido()***: Devuelve el nombre del ganador que ha alcanzado una mayor velocidad media en la edición en la que ha sido ganador. Si no se puede calcular devuelve null.
- ***Map<String, List<Ganador>> getGanadoresPorNacionalidad()***: Devuelve un Map en el que las claves son las nacionalidades y los valores la lista de ganadores de esa nacionalidad.
- ***Map<String, Long> cuentaGanadoresPorNacionalidad()***: Devuelve un Map en el que las claves son las nacionalidades y los valores el número de ganadores de esa nacionalidad.
- ***Boolean hayAlgunGanador(String nacionalidad)***: Devuelve *true* si entre los ganadores de la carrera hay alguno de la nacionalidad dada como parámetro.