

Ejercicio: Ciclismo

Autor: Mariano González. **Revisores:** José Antonio Troyano, José Riquelme, Carlos G. Vallejo, Fermín Cruz. **Última modificación:** 28/5/2018

En este ejercicio vamos a trabajar con información sobre los ciclistas que han ganado una prueba ciclista por etapas, como la *Vuelta a España*, el *Giro de Italia* o el *Tour de France*. Para ello disponemos de datos obtenidos de la web oficial de la carrera (por ejemplo, www.letour.fr para el *Tour de France*).

Para cada ganador de la carrera se tiene el año en que la ganó, su nacionalidad, su nombre, su equipo, el número total de kilómetros recorridos en la carrera, el tiempo total empleado para hacer el recorrido, el número de etapas ganadas y el número de días que vistió el maillot de líder de la carrera.

Los datos se encuentran almacenados en ficheros en formato CSV codificados en UTF-8. Cada registro de un fichero ocupa una línea y contiene los datos correspondientes al ganador de una edición de la carrera. Por ejemplo, estas son algunas líneas del fichero con los ganadores del *Tour de France*:

```
2013,Great Britain,Chris Froome,Team Sky,3404,PT83H56M20S,3,14
2014,Italy,Vincenzo Nibali,Astana,3661,PT89H59M6S,4,19
2015,Great Britain,Chris Froome,Team Sky,3360,PT84H46M14S,1,16
2016,Great Britain,Chris Froome,Team Sky,3529,PT89H4M48S,2,14
2017,Great Britain,Chris Froome,Team Sky,3540,PT86H20M55S,0,15
```

En primer lugar diseñaremos los tipos que intervienen en el sistema. A continuación leeremos los datos de los ganadores desde un fichero CSV. Luego realizaremos algunas operaciones con los datos.

Antes de comenzar, descarga e importa en Eclipse el proyecto con las interfaces, las clases de test, las clases de utilidad y los datos de los ganadores que necesitas para trabajar.

1. Definición de los tipos

Vamos a diseñar los tipos que necesitamos. Estos tipos son **Ganador**, que representa el ganador de una edición de una carrera, y **EstadisticasCarrera**, que agrupa los ganadores de las distintas ediciones de una carrera y permite obtener estadísticas sobre ella.

1.1 El tipo Ganador

Este tipo nos permitirá representar el ganador de una edición de la carrera. Un ganador tiene un **año**, de tipo Integer; una **nacionalidad**, de tipo String; un **nombre**, de tipo String; un **equipo**, de tipo String; unos **kilómetros recorridos**, de tipo Integer; un **tiempo empleado**, de tipo Duration; un **número de etapas ganadas**, de tipo Integer; un **número de días con maillot de líder**, de tipo Integer, y una **velocidad media**, de tipo Double. Todas las propiedades son consultables.

EJERCICIO 1: Implementa el tipo Ganador, partiendo de la interfaz disponible en el proyecto y teniendo en cuenta lo siguiente:

- La clase tendrá un constructor que recibirá un parámetro por cada propiedad básica del tipo, en el orden en que están definidas.
- Tendrá también un constructor a partir de String. La cadena de entrada contendrá, separados por comas, el año, la nacionalidad, el nombre, el equipo, los kilómetros recorridos, el tiempo empleado, el número de etapas ganadas y el número de días con el maillot de líder. Un ejemplo de cadena de entrada sería el siguiente:
`"1995,Spain,Miguel Indurain,Banesto,3653,PT92H44M59S,2,12"`
- El tipo Ganador tiene las siguientes restricciones:
 - El número de kilómetros recorridos debe ser mayor que cero
 - El número de etapas ganadas debe ser mayor o igual que cero.
 - El número de días con el maillot de líder debe ser mayor o igual que cero.
 - El tiempo empleado debe ser mayor que cero. Utiliza la constante de tipo `Duration.DURATION_ZERO`.
- Dos ganadores serán iguales si tienen el mismo año, el mismo nombre y el mismo equipo.
- La representación como cadena incluirá el año, el nombre y el equipo.

Prueba la implementación ejecutando la clase `TestGanador` del proyecto. Al ejecutar el test, debe mostrarse en pantalla el siguiente resultado:

```
Ganador de la carrera: GanadorImpl [anyo=1995, nombre=Miguel Indurain,
equipo=Banesto]
Tiempo empleado: PT92H44M59S
```

1.2 El tipo `EstadisticasCarrera`

Este tipo representa las estadísticas de una carrera. Tiene como propiedades un **nombre** de carrera, de tipo String; un **número de ediciones**, de tipo Integer, y los **ganadores** de la carrera, de tipo `List<Ganador>`. Las dos primeras propiedades son consultables.

EJERCICIO 2: Implementa el tipo `EstadisticasCarrera`, partiendo de la interfaz disponible en el proyecto y teniendo en cuenta lo siguiente:

- La clase tendrá un constructor que recibirá un parámetro por cada propiedad básica del tipo, en el orden en que están definidas.
- Tendrá también un constructor que recibirá como parámetros el nombre de una carrera y el nombre de un fichero CSV con datos de los ganadores de la carrera. El constructor creará un objeto `EstadisticasCarrera` con todos los ganadores contenidos en el fichero CSV. Para ello, se apoyará en el método `leeFichero` de la clase de utilidad `Ficheros` incluida en el proyecto.
- La representación como cadena será el nombre de la carrera.

2. Lectura de datos

Los datos de los ganadores se encuentran en los ficheros `ganadores_Carrera.csv` situados dentro de la carpeta `datos` del proyecto (siendo `Carrera` el nombre abreviado de la carrera). Para leer estos datos usaremos el constructor de la clase `EstadisticasCarreraImpl`, que como vimos antes construye unas estadísticas a partir de un fichero CSV.

EJERCICIO 3: Completa la clase `TestEstadisticasCarrera` del proyecto para que cree unas estadísticas de nombre “Tour de France” con los ganadores que hay en el fichero `ganadores_Tour.csv` situado dentro de la carpeta `datos`. Una vez creadas las estadísticas, muestra en la pantalla el nombre de la carrera y el número de ediciones. Al ejecutar el test, debe mostrarse en pantalla el siguiente resultado:

```
Carrera: Tour de France
Número de ediciones: 30
```

3. Operaciones de consulta

Queremos disponer de las siguientes operaciones en el tipo `EstadisticasCarrera`:

1. `List<String> getGanadoresConRecorridoInferiorA(Integer km)`: devuelve una lista con los nombres de los ganadores que han recorrido menos kilómetros que los dados como parámetro.
2. `Boolean hanGanadoTodosAlgunaEtapa()`: devuelve true si todos los ganadores han ganado alguna etapa en la edición en la que ganaron la carrera.
3. `Long getNumeroGanadores()`: devuelve el número de ganadores distintos de la carrera. Si un ciclista ha ganado la carrera en varias ediciones, solo se debe contar una vez.
4. `Set<String> getEquiposGanadores()`: devuelve un conjunto con los nombres de los equipos que han ganado la carrera.
5. `Ganador buscaGanador(String nombre)`: devuelve el primer ganador que concuerde con el nombre dado como parámetro. Si no se encuentra ninguno, se devuelve null.
6. `Ganador buscaGanador(Integer anyo)`: devuelve el ciclista ganador de ese año, si es que existe. Si no se encuentra ninguno, se devuelve null.
7. `Integer calculaDistanciaTotal()`: devuelve la suma de los kilómetros recorridos en todas las ediciones de la carrera.
8. `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.
9. `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.
10. `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.
11. `Map<String, List<Ganador>> getGanadoresPorNacionalidad()`: devuelve un Map en el que las claves son las nacionalidades y los valores la lista de ganadores de cada nacionalidad.

12. *Map<String, Long> cuentaGanadoresPorNacionalidad()*: devuelve un Map en el que las claves son las nacionalidades y los valores el número de ganadores de cada nacionalidad.
13. *Map<String, Integer> getTotalEtapasGanadasPorEquipo()*: devuelve un Map en el que las claves son los equipos y los valores el número de etapas ganadas por cada equipo.
14. *Boolean hayAlgunGanadorNacionalidad(String nacionalidad)*: devuelve true si entre los ganadores de la carrera hay alguno de la nacionalidad dada como parámetro.
15. *void escribeGanadoresNacionalidadConAnyos(String nombreFichero, String nacionalidad)*: toma como parámetros el nombre de un fichero, y una nacionalidad. Genera un fichero con los ganadores de esa nacionalidad y los años en los que ganaron la carrera. Ejemplo del fichero generado:

```
Carlos Sastre -> [2008]
Alberto Contador -> [2007, 2009]
Miguel Indurain -> [1991, 1992, 1993, 1994, 1995]
Oscar Pereiro -> [2006]
```

16. *void escribeGanadoresDeAnyosOrdenados(String nombreFichero, Integer anyoInicial, Integer anyoFinal)*: toma como parámetros el nombre de un fichero, el año inicial y el año final del intervalo temporal [anyoInicial, anyoFinal). Genera un fichero con el nombre dado como parámetro con los ciclistas ganadores de la carrera. En el fichero solamente deben aparecer los ganadores que ganaron en los años que van de anyoInicial (inclusive) hasta anyoFinal (sin incluirlo). Además, el orden en el que aparecen los ciclistas es de ganador más reciente a ganador más lejano en el tiempo. Ejemplo del fichero generado:

```
2007 - Alberto Contador (Spain)
2006 - Oscar Pereiro (Spain)
```

17. *Long getNumeroNacionalidadesGanadoresUltimosAños(Integer n)*: dado un número n devuelve el número de nacionalidades distintas que han ganado la carrera en los últimos n años.
18. *Map<String, Integer> getNumeroEtapasLiderPorEquipo()*: devuelve un Map que relaciona cada equipo con el número total de etapas que han sido líderes los ganadores de ese equipo.
19. *void muestraEdiciones()*: muestra en la consola un listado con los años de cada edición y el número de kilómetros recorridos ordenados por kilómetros de mayor a menor
20. *String getNacionalidadMasGanadores()*: devuelve la nacionalidad con mayor número de ganadores. Utiliza el método cuentaGanadoresPorNacionalidad.

EJERCICIO 4: Escribe el código de cada método en la clase EstadisticasCarreraImpl. Prueba cada uno de ellos añadiendo una llamada al mismo en la clase TestEstadisticasCarrera.