

# Desarrollo de Software - Práctica 1.

## Ejercicio 1 - Patrón Factoría Abstracta.

Adrián Portillo Sánchez

Para el ejercicio 1, el cual pedía implementar dos carreras simultáneas de bicicletas, he utilizado el patrón factoría abstracta.

Para implementar la jerarquía de bicicletas en primer lugar he implementado la clase abstracta `Bicicleta`, la cual se implementa en dos clases específicas `BicicletaCarretera` y `BicicletaMontaña`, y posee una serie de piezas de bicicletas.

Para implementar las piezas de las bicicletas es para lo que he utilizado la primera factoría, en primer lugar he implementado la jerarquía de piezas de bicicleta la cual se compone de tres clases abstractas `Manillar`, `Cuadro` y `Rueda` cada una especificada en otras dos clases según la pieza es de montaña o de carretera, y para crear las piezas he creado una factoría abstracta `FactoriaPiezas`, la cual posee tres métodos: `crearManillar`, `crearCuadro` y `crearRueda`; que crearán una pieza de montaña o de carretera según el parámetro que se le pase.

Una vez implementada la jerarquía de bicicletas, la segunda factoría la he utilizado para crear los arrays de `Bicicleta`; para ello he creado una interfaz `Carrera` la cual implementa dos clases: `FactoriaCarreraCarretera` y `FactoriaCarreraMontaña`; ambas crear un array de `n` bicicletas de carrera o de montaña según la factoría utilizada.

Para lanzar las hebras creo dos clases que heredan de `Thread`: `CarreraMontaña` y `CarreraCarretera`, cada una de ellas utilizando la `FactoriaCarreras`, posee un array de bicicletas del tipo correspondiente; en el método `run()` se esperan 60 segundos y se hace el método `shuffle()` sobre los arrays, generando un resultado aleatorio en el cual el primer elemento es el ganador y los `n` últimos elementos se retiran, poniendo su referencia a `null`, siendo `n` un 10 % de los corredores en carreras de carretera y el 20 % en carreras de montaña. El método `main()` de principal simplemente instancia las dos carreras con un número aleatorio de corredores y ejecuta su método `run()`.

Se crearán carreras de 31 corredores.

Se ha iniciado la carrera de Montaña  
Se ha iniciado la carrera de Carretera  
Se retira el corredor 30 de la carrera de montaña.  
Se retira el corredor 12 de la carrera de montaña.  
Se retira el corredor 4 de la carrera de montaña.  
Se retira el corredor 31 de la carrera de montaña.  
Se retira el corredor 9 de la carrera de montaña.  
Se retira el corredor 44 de la carrera de carretera.  
Se retira el corredor 56 de la carrera de carretera.  
Se retira el corredor 40 de la carrera de carretera.  
Se retira el corredor 3 de la carrera de montaña.  
Se han retirado 6 corredores de la carrera de montaña.  
Se han retirado 3 corredores de la carrera de carretera.

El ganador de la carrera de carretera es el corredor con la bicicleta número 43.

El ganador de la carrera de montaña es el corredor con la bicicleta número 21.

Figura 1: Ejemplo de una ejecución del programa.

A continuación se muestra el diagrama de clases del programa donde podemos ver de una forma más gráfica esto que he descrito, por cuestiones de visualización también dejo una copia en formato .png junto a este documento.

