

MOD06 LAB01 – Métodos y encapsulamiento

1. Se deberá crear un paquete llamado `com.training.java.basic.capitulo6`.
2. **STATIC** – Se crea una clase `StaticTest` con un entero estático y otro “normal”, y dos métodos que incrementen cada uno de ellos y los saque por pantalla. Se deberá crear un método `main` en esa misma clase que cree varios objetos `StaticTest`. La finalidad de este ejercicio es ir jugando con los objetos, llamando a los métodos de incremento de cada uno de ellos una o varias veces.
3. **CLASES con ATRIBUTOS** – Se crea una clase `Ciclista` y se definen en ella unos cuantos atributos: nombre, ciudad, calle, número, usa casco.
4. Se crea una clase `Luz` con los atributos color, estado, nivel de batería.
5. Se crea una clase `Bicicleta` con los atributos color, número de marchas, velocidad actual, marcha actual.
6. **CONSTRUCTORES** – Se añade a cada clase un constructor que reciba todos los datos obligatorios para crear sus objetos.
7. **ATRIBUTOS** – A los atributos de las bicicletas, se deberá añadir una luz delantera y una luz trasera. Y a los ciclistas, se deberá añadir una lista de bicicletas.
8. **SETTERS y GETTERS** – Se añaden a cada clase métodos para acceder a los valores de sus atributos.
9. **MÉTODOS** - ¿Qué acciones se pueden hacer sobre cada uno de estos objetos? Se deberán crear métodos para apagar y encender las luces, arrancar, acelerar, frenar, subir y bajar de marcha, comprar y vender una bici.
 1. Los métodos que enciendan y apaguen las luces tendrán dos variantes: para el ciclista este método seleccionará la bici (de su colección) de la cual apagar/encender las luces. En la clase `bicicleta` estos métodos apagarán o encenderán las luces; es decir, cambiarán su estado lógico.
 2. Arrancar la bici pondrá la marcha actual y la velocidad a 1.
 3. Acelerar incrementará la velocidad actual.
 4. Frenar disminuirá la velocidad actual.
 5. Subir/bajar de marcha aumentará o disminuirá la marcha actual.
 6. Comprar/vender bici añadirá o borrará una bici de la lista de bicis en posesión del ciclista.

Para implementar los métodos habrá que usar los getters y setters creados anteriormente para modificar el estado de las variables, también se deberán imprimir por pantalla los cambios realizados. (ej: `subirMarcha()` imprimirá un mensaje parecido a “Se ha subido la marcha”.)
10. **OBJETOS** – Se deberá escribir un programa en el que se declare un ciclista con su bici.
11. **MENSAJES** – Ese ciclista deberá encender las luces de su bici, arrancar, acelerar a 10 km/h y usar la segunda marcha. Se implementará un método *main* en una nueva clase que realice las diversas acciones imprimiendo por pantalla los pasos realizados por el ciclista.