Reto 1 – Alquiler de Bicicletas

Objetivo:

El objetivo de este reto es que el estudiante reconozca y aplique los elementos básicos del paradigma de la programación orientada a objetos en un escenario abstraído de la cotidianidad.

Contexto:

Una empresa busca posicionarse en el mercado de una ciudad con un sistema automático que le permita a cualquier persona alquilar o rentar una bicicleta por medio de una aplicación móvil. Para mantener un control sobre el servicio que está prestando la empresa ha definido que una bicicleta que se encuentra alquilada en un momento dado está “En Servicio”.

En el sistema actual una **Bicicleta** se define como se muestra en el diagrama:

Bicicleta
- Id: String - EnServicio: boolean
+ getEnServicio(): boolean + getId(): String + BicicletasEnServicio(Bicicletas[] bicicletas): Bicicleta[] + BicicletasDisponibles(Bicicletas[] bicicletas): Bicicleta[]

Reto:

1. Las funciones **getId** y **getEnServicio** retornan el Id y el valor de EnServicio respectivamente.
2. La función **BicicletasEnServicio** toma como argumento un arreglo de tipo Bicicleta y retorna un **arreglo** con las bicicletas que están en servicio. Del mismo modo la función **BicicletasDisponibles** retorna un arreglo con las bicicletas que NO están en servicio.

Cabe destacar que el arreglo que retornan estas dos funciones debe tener un tamaño igual a la cantidad de bicicletas que cumplen la condición encontradas. Es decir, si en la función **BicicletasDisponibles** se encuentran 6 bicicletas que no están en servicio, el arreglo retornado debe ser de tamaño 6, ni más ni menos.

Casos de Prueba:

Para validar el correcto funcionamiento del programa considere los siguientes escenarios:

Datos de Entrada:

Para un arreglo como el siguiente:

0	1	2	3	4	5
Id: Bicicleta1 EnServicio: false	Id: Bycicle2 EnServicio: false	Id: Bicyclette3 EnServicio: true	Id: Fahrrad4 EnServicio: false	Id: Velosipyed5 EnServicio: true	Id: Biciklo6 EnServicio: false

<table>
<tr><th>Caso de Prueba</th><th>Salida Esperada</th></tr>
<tr><td>1. BicicletasEnServicio()</td><td>
<table>
<tr><th>0</th><th>1</th></tr>
<tr><td>Id: Bicyclette3
EnServicio: true</td><td>Id: Velosipyed5
EnServicio: true</td></tr>
</table>
</td></tr>
<tr><td>2. BicicletasDisponibles()</td><td>
<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>Id: Bicicleta1
EnServicio: false</td><td>Id: Bycicle2
EnServicio: false</td><td>Id: Fahrrad4
EnServicio: false</td><td>Id: Biciklo6
EnServicio: false</td></tr>
</table>
</td></tr>
</table>

ENTREGA:

1. El archivo que suba a la plataforma para su calificación debe llamarse **exactamente** "Bicicleta.java", de lo contrario no se calificará.
2. Los nombres de las clases, miembros dato y funciones deben llamarse **exactamente** como se muestra en el diagrama mostrado al comienzo del reto, la firma de su clase debe ser cómo se muestra en la siguiente imagen:

```java
public class Bicicleta {

    private String Id;
    private boolean EnServicio;

    public Bicicleta(String _Id, boolean _EnServicio){
        //Implementación
    }

    public String getId() {
        //Implementación
    }

    public boolean getEnServicio(){
        //Implementación
    }

    public static Bicicleta[] BicicletasEnServicio(Bicicleta[] bicicletas){
        //Implementación
    }

    public static Bicicleta[] BicicletasDisponibles(Bicicleta[] bicicletas){
        //Implementación
    }
}
```