Estación de BiciPalma

Vamos a construir un prototipo de la aplicación que gestiona una estación donde se anclan las bicicletas del servicio BiciPalma.

1. Crea un proyecto con tu nombre y apellidos.
2. Copia y pega la función principal BiciPalma.java. Utilízala como guía en el proceso TDD.
3. Completa las clases que necesites implementando los casos test que se proponen en la función principal. No puedes modificar su código.
4. Organiza las clases en sus paquetes correspondientes, por si necesitas construir una función main() para testear cada clase por separado.

### **Cómo entregar el código**

1. Exporta el proyecto con la utilidad adecuada de Eclipse.
2. Envíame el archivo por correo electrónico.

### **Clase Estacion**

#### **Atributos**

* id = identificador de la estación
* direccion = dirección de la estación
* numeroAnclajes = número de anclajes de la estación
* anclajes[numeroAnclajes] = array donde almacenarás las bicicletas que se anclen en la estación

#### **Métodos**

* consultarEstacion() muestra un mensaje con id, direccion y numeroAnclajes
* anclajesLibres() devuelve la cantidad de anclajes libres
* consultarAnclajes() recorre el array anclajes y muestra el id de la bici anclada o si está libre
* anclarBicicleta(bicicleta) inserta el objeto bicicleta en el primer registro libre del array anclajes y llama a mostrarAnclaje()
* mostrarAnclaje(bicicleta, numeroAnclaje) muestra un mensaje con el id de la bicicleta anclada y en qué anclaje se ha anclado
* leerTarjetaUsuario(tarjetaUsuario) comprueba si la tarjeta de usuario está activada
* retirarBicicleta(tarjetaUsuario)
  + si la tarjeta de usuario está activada retira una bici del anclaje y muestra su información en pantalla llamando a mostrarBicicleta(bicicleta, numeroAnclaje)
  + para simular que un usuario/a del servicio ancla una bici, debes generar una posición al azar entre los anclajes y eliminar esa bici con el método generarAnclaje()
  + para eliminar una bici del array debes poner esa posicion del array a null.
* mostrarBicicleta(bicicleta, numeroAnclaje) muestra un mensaje con el id de la bici y el número de anclaje donde estaba.
* generarAnclaje() devuelve una posición al azar entre el número de anclajes. Utiliza ThreadLocalRandom.current()

### **Clase Bicicleta**

#### **Atributos**

* id = identificador de la bicicleta: un número de tres dígitos.

### **Clase TarjetaUsuario**

#### **Atributos**

* id = identificador de la tarjeta: un número de nueve dígitos.
* activada = valor lógico que indica si la tarjeta está activada o no.

### **Solución**

<https://github.com/dfleta/bicipalma>