

JUnit Concesionario

Jaime Herrerias Jimenez

ConcesionarioTest

Empezamos probando el concesionario Test creamos una nueva clase para tal fin. Una vez la tengamos creada empezamos con nuestras pruebas. En primer lugar probamos a añadir varios coches correctos, probando matricula sin espacio, con guion y con espacios. Una vez comprobado nos vamos a ver los posibles casos erróneos: matricula con más números o más letras, faltando numero o letras, matriculas en minúsculas etc

```

10
11 @Test
12 public void testAnnadirMatricula() {
13     // coche correcto
14     assertTrue(concesionario.annadir("1234GGG", Color.AZUL, Modelo.SERIE5));
15     assertTrue(concesionario.annadir("9999GGH", Color.AZUL, Modelo.SERIE5));
16     assertTrue(concesionario.annadir("1234-FFF", Color.AZUL, Modelo.SERIE5));
17     assertTrue(concesionario.annadir("1234 GFF", Color.AZUL, Modelo.SERIE5));
18     // coche repetido
19     assertFalse(concesionario.annadir("1234GGG", Color.AZUL, Modelo.SERIE5));
20     assertFalse(concesionario.annadir("1234-FFF", Color.AZUL, Modelo.SERIE5));
21     assertFalse(concesionario.annadir("1234 GFF", Color.AZUL, Modelo.SERIE5));
22     // Sobran numeros
23     assertFalse(concesionario.annadir("143348BBB", Color.ROJO, Modelo.SERIE1));
24     // Sobran letras
25     assertFalse(concesionario.annadir("12348DDD", Color.ROJO, Modelo.SERIE1));
26     // falta un numero
27     assertFalse(concesionario.annadir("123BBB", Color.ROJO, Modelo.SERIE1));
28     // falta una letra
29     assertFalse(concesionario.annadir("1234BB", Color.ROJO, Modelo.SERIE1));
30     // letra minuscula
31     assertFalse(concesionario.annadir("1234BBb", Color.ROJO, Modelo.SERIE1));
32     // numero entre las letras
33     assertFalse(concesionario.annadir("143481B", Color.ROJO, Modelo.SERIE1));
34     // letra entre los numeros
35     assertFalse(concesionario.annadir("1G348BBB", Color.ROJO, Modelo.SERIE1));
36     // todo numero
37     assertFalse(concesionario.annadir("0000000", Color.AZUL, Modelo.SERIE5));
38     // todo letras
39     assertFalse(concesionario.annadir("FFFFFFF", Color.AZUL, Modelo.SERIE5));
40     // caracter erroneo
41     assertFalse(concesionario.annadir("1G34*BBB", Color.ROJO, Modelo.SERIE1));
42
43 }
44

```

Probamos los diferentes posibilidades de colores y nulo:

```

44
45 @Test
46 public void testAnnadirColor() {
47     // Colores correctos
48     assertTrue(concesionario.annadir("5555FFF", Color.AZUL, Modelo.SERIE5));
49     assertTrue(concesionario.annadir("2333FTF", Color.ROJO, Modelo.SERIE5));
50     assertTrue(concesionario.annadir("3343FTF", Color.PLATA, Modelo.SERIE5));
51     // color null
52     assertFalse(concesionario.annadir("3232FTT", null, Modelo.SERIE5));
53
54 }
55

```

Y comprobamos que el valor que devuelve los diferentes colores no es nulo.

```

56
57 @Test
58 public void probandoColoresValidos() {
59     //comprobamos que no es nulo lo que devuelve
60     assertNotNull(concesionario.getCochesColor(Color.PLATA));
61     assertNotNull(concesionario.getCochesColor(Color.ROJO));
62     assertNotNull(concesionario.getCochesColor(Color.AZUL));
63 }
64

```

Probamos todos los posibles modelos y un modelo nulo:

```

65 @Test
66 public void testAnnadirModelo() {
67     // Colores correctos
68     assertTrue(concesionario.annadir("5133DDD", Color.AZUL, Modelo.SERIE1));
69     assertTrue(concesionario.annadir("2233FFF", Color.AZUL, Modelo.SERIE2));
70     assertTrue(concesionario.annadir("7333TTF", Color.AZUL, Modelo.SERIE3));
71     assertTrue(concesionario.annadir("8433FFF", Color.AZUL, Modelo.SERIE5));
72     assertTrue(concesionario.annadir("9533FFF", Color.AZUL, Modelo.TOLEDO));
73     assertTrue(concesionario.annadir("0633FFF", Color.AZUL, Modelo.CORDOBA));
74     assertTrue(concesionario.annadir("1733FFF", Color.AZUL, Modelo.IBIZA));
75     // color null
76     assertFalse(concesionario.annadir("4783GTR", Color.AZUL, null));
77 }
78

```

Vamos a ver como eliminamos coches, en primer lugar añadimos cuatro coches hacemos un size() y también lo comprobamos. Vemos que como correctamente indicamos hay cuatro coches creados. Probamos a eliminarlo hacemos un size(), y vemos que todo ha ido correcto. También comprobamos que no se puede eliminar coches inexistentes por lo que vamos a probar a eliminar el coche anterior que ya fue eliminado y vemos que es correcto.

```

80 @Test
81 public void testEliminar() {
82     // creamos algunos coches para las posteriores pruebas
83     concesionario.annadir("0125TRT", Color.AZUL, Modelo.TOLEDO);
84     concesionario.annadir("0816BBB", Color.AZUL, Modelo.SERIE5);
85     concesionario.annadir("0833BBB", Color.ROJO, Modelo.SERIE1);
86     concesionario.annadir("9933BBB", Color.AZUL, Modelo.SERIE2);
87     // Comprobamos que quedan 4 coches en el array
88     assertEquals(4, concesionario.size());
89     // eliminamos
90     assertTrue(concesionario.eliminar("0816BBB"));
91     // Comprobamos que quedan 3 coches en el array
92     assertEquals(3, concesionario.size());
93     // eliminamos un coche que ya no existe
94     assertFalse(concesionario.eliminar("0816BBB"));
95     // Comprobamos que quedan 3 coches en el array
96     assertEquals(3, concesionario.size());
97 }
98

```

CocheTest

Y hacemos la misma operación con cocheTest, agregamos varios coches y probamos que sean correctos, ahora metemos errores como por ejemplo que sobren letras números o haya caracteres inválidos.

```

29     public void testInstanciarCocheStringColorModelo() {
30         //COCHES VALIDOS
31         assertNotNull(Coche.instanciarCoche("0819BMR", Color.AZUL, Modelo.CORDOBA));
32         assertNotNull(Coche.instanciarCoche("1234-GBX", Color.PLATA, Modelo.SERIE1));
33         assertNotNull(Coche.instanciarCoche("1234 FTG", Color.ROJO, Modelo.SERIES));
34
35         //falta una letra
36         assertNull(Coche.instanciarCoche("0819BR", Color.AZUL, Modelo.CORDOBA));
37         //falta numero
38         assertNull(Coche.instanciarCoche("089BMR", Color.AZUL, Modelo.CORDOBA));
39         //sobra numero
40         assertNull(Coche.instanciarCoche("1260FFFF", Color.ROJO, Modelo.SERIES));
41         //sobra letra
42         assertNull(Coche.instanciarCoche("1260FFFF", Color.ROJO, Modelo.SERIES));
43         //caracter invalido
44         assertNull(Coche.instanciarCoche("0819_BMR", Color.AZUL, Modelo.CORDOBA));
45         assertNull(Coche.instanciarCoche("1459sbmr", Color.AZUL, Modelo.CORDOBA));
46         //todo letras
47         assertNull(Coche.instanciarCoche("0819123", Color.AZUL, Modelo.CORDOBA));
48         //todo numeros
49         assertNull(Coche.instanciarCoche("DDDDBMR", Color.AZUL, Modelo.CORDOBA));
50         //modelo nulo
51         assertNull(Coche.instanciarCoche("1234GBX", Color.PLATA, null));
52         //color nulo
53         assertNull(Coche.instanciarCoche("0819BMR", null, Modelo.CORDOBA));
54         //digitos invieridos
55         assertNull(Coche.instanciarCoche("FTG1234", Color.ROJO, Modelo.SERIES));
56
57     }
58

```

Y lo mismo con la matricula:

```

60     @Test
61     public void testInstanciarCocheString() {
62         //valido
63         assertNotNull(Coche.instanciarCoche("1234FFF"));
64         assertNotNull(Coche.instanciarCoche("1234-HBD"));
65         assertNotNull(Coche.instanciarCoche("1234 FGG"));
66         //falta una letra
67         assertNull(Coche.instanciarCoche("0819BR"));
68         //falta numero
69         assertNull(Coche.instanciarCoche("089BMR"));
70         //sobra numero
71         assertNull(Coche.instanciarCoche("1260FFF"));
72         //sobra letra
73         assertNull(Coche.instanciarCoche("1260FFFF"));
74         //caracter invalido
75         assertNull(Coche.instanciarCoche("0819_BMR"));
76         assertNull(Coche.instanciarCoche("1459sbmr"));
77         //todo letras
78         assertNull(Coche.instanciarCoche("0819123"));
79         //todo numeros
80         assertNull(Coche.instanciarCoche("DDDDBMR"));
81
82     }
83

```