

```
1 /**
2  package week3Tests;
3
4  import static org.junit.jupiter.api.Assertions.*;
5
6  /**
7   * @author patrickF
8   *
9   */
10 class CarTest {
11     Car car1;
12     Car car2;
13     Name owner1;
14
15     /**
16      * @throws java.lang.Exception
17      */
18     @BeforeEach
19     void setUp() throws Exception {
20         // Create an owner object & set the name attributes only ...no cars have
21         been
22         // linked to this owner as yet
23         owner1 = new Name();
24         owner1.setFirstName("Alan");
25         owner1.setMiddleName("Mathson");
26         owner1.setSurname("Turing");
27
28         // Create a car and link ut to owner1.
29         car1 = new Car("Mercedes Benz", 200, owner1, 0.33);
30
31         // Create an array of cars whose size is the same as the array of cars in
32         the
33         // Name class
34         int carArraySize = owner1.getOwnedCars().length;
35         Car[] ownedCars = new Car[carArraySize];
36
37         // Filling the array with the data of the car created.
38         ownedCars[0] = car1; // store the car into the first location of the array
39         of 1
40
41         // Set the array of ownedCars as the array of car owned by the Name object
42         owner1.setOwnedCars(ownedCars);
43     }
44
45     /**
46      * @throws java.lang.Exception
47      */
48     @AfterEach
49     void tearDown() throws Exception {
50         car1 = null;
51         owner1 = null;
52     }
53
54     // Testing Getters from car1 object.
55     @Test
56     void testGetModel() {
57         assertEquals("Mercedes Benz", car1.getModel());
58     }
59 }
```

```
67     @Test
68     void testGetTankSize() {
69         assertEquals(200, car1.getTankSize());
70     }
71
72     @Test
73     void testGetManfMPG() {
74         assertEquals(0.33, car1.getManfMPG());
75     }
76
77     @Test
78     void testGetOwnerName() {
79         assertEquals("Alan", car1.getOwnerName().getFirstName());
80     }
81
82     // Setters from car1 object.
83     @Test
84     void testSetModel() {
85         car1.setModel("NWMDL");
86         assertTrue(car1.getModel() == "NWMDL");
87     }
88
89     @Test
90     void testSetTankSize() {
91         car1.setTankSize(33);
92         assertEquals(33, car1.getTankSize());
93     }
94
95     @Test
96     void testSetManfMPG() {
97         car1.setManfMPG(12.55);
98         assertEquals(12.55, car1.getManfMPG());
99     }
100
101     @Test
102     void testSetOwnerName() {
103         Name owner2 = new Name();
104         owner2.setFirstName("Patrick");
105         owner2.setMiddleName("Joseph");
106         owner2.setSurname("Frendo");
107
108         car2 = new Car("Ford Fiesta", 450, owner2, 0.33);
109
110         int carArraySize = owner1.getOwnedCars().length;
111         Car[] ownedCars1 = new Car[carArraySize];
112
113         ownedCars1[0] = car2;
114
115         car2.setOwnerName(owner2);
116         assertEquals("Patrick", car2.getOwnerName().getFirstName());
117         owner2 = null;
118         car2 = null;
119     }
120
121     // Constructors
122     @Test
123     void testOverloadedConstructorCar() {
124         Name owner3 = new Name();
125         owner3.setFirstName("Peter");
126         owner3.setMiddleName("Paul");
```

```
127         owner3.setSurname("Vandike");
128
129         Car carx = new Car("Clown Car", 800, owner3, 2);
130
131         int carArraySize = owner3.getOwnedCars().length;
132         Car[] ownedCars = new Car[carArraySize];
133         carx.setOwnerName(owner3);
134         assertEquals("Clown Car", carx.getModel());
135         assertEquals(800, carx.getTankSize());
136         assertEquals("Peter Paul Vandike", carx.getOwnerName().getFullName());
137
138
139     }
140
141     // Methods
142     @Test
143     void testTankBigger_BiggerThanTankSize() {
144         assertEquals("Inefficient fuel use", car1.tankBigger(150));
145     }
146
147     @Test
148     void testTankBigger_SmallerThanTankSize() {
149         assertEquals("Good fuel consumption", car1.tankBigger(750));
150     }
151
152     @Test
153     void testTankBigger_SameTankSize() {
154         assertEquals("Average consumer", car1.tankBigger(200));
155     }
156
157     @Test
158     void testGetOwner() {
159         assertEquals("Alan Mathson Turing", car1.getOwner());
160     }
161
162     @Test
163     void testEstimateDistance() {
164         assertEquals(14.52, car1.estimateDistance());
165     }
166
167 }
168
```