

```
1 /**
2  *
3  */
4 package week3Tests;
5
6 import static org.junit.jupiter.api.Assertions.*;
7
8
9
10
11
12
13
14
15 /**
16  * @author patrickF
17  *
18  */
19 class NameTest {
20     Name owner1;
21     Car car1;
22     Car car2;
23     Car car3;
24     Car car4;
25     Car car5;
26
27     /**
28      * @throws java.lang.Exception
29      */
30     @BeforeEach
31     void setUp() throws Exception {
32         // Create an owner object & set the name attributes only ...no cars have
33         been
34         // linked to this owner as yet
35         owner1 = new Name();
36         owner1.setFirstName("Alan");
37         owner1.setMiddleName("Mathson");
38         owner1.setSurname("Turing");
39
40         // Create a set of 5 cars and will link them to owner1.
41         car1 = new Car("Mercedes Benz", 200, owner1, 0.33);
42         car2 = new Car("Toyota Starlet", 150, owner1, 0.23);
43         car3 = new Car("Mitsubishi Colt", 100, owner1, 0.88);
44         car4 = new Car("MiniCooper", 200, owner1, 0.33);
45         car5 = new Car("Ford Fiesta", 450, owner1, 0.33);
46
47         // Create an array of cars whose size is the same as the array of cars in
48         the
49         // Name class
50         int carArraySize = owner1.getOwnedCars().length;
51         Car[] ownedCars = new Car[carArraySize];
52
53         // Filling the array with the data of the cars created.
54         ownedCars[0] = car1; // store the first car into the first location of the
55         array
56         ownedCars[1] = car2;
57         ownedCars[2] = car3;
58         ownedCars[3] = car4;
59         ownedCars[4] = car5;
60
61         // Set the array of ownedCars as the array of cars owned by the Name object
62         owner1.setOwnedCars(ownedCars);
63         // Car[] list = owner1.getOwnedCars();
64     }
65
66     /**
67      * @throws java.lang.Exception
68      */
69     @AfterEach
```

```
67     void tearDown() throws Exception {
68         owner1 = null;
69         car1 = null;
70         car2 = null;
71         car3 = null;
72         car4 = null;
73         car5 = null;
74     }
75
76     // Testing Getters from owner1 object.
77     @Test
78     void testGetFirstName() {
79         assertEquals("Alan", owner1.getFirstName());
80     }
81
82     @Test
83     void testGetMiddleName() {
84         assertEquals("Mathson", owner1.getMiddleName());
85     }
86
87     @Test
88     void testGetSurname() {
89         assertEquals("Turing", owner1.getSurname());
90     }
91
92     @Test
93     void testGetOwnedCars() {
94         Car[] list = owner1.getOwnedCars();
95         assertEquals("Mitsubishi Colt", list[2].getModel());
96     }
97
98     // Setters of object owner1
99     @Test
100    void testSetFirstName() {
101        owner1.setFirstName("Cuddles");
102        assertEquals("Cuddles", owner1.getFirstName());
103    }
104
105    @Test
106    void testSetMiddleName() {
107        owner1.setMiddleName("Midcuddles");
108        assertEquals("Midcuddles", owner1.getMiddleName());
109    }
110
111    @Test
112    void testSetSurname() {
113        owner1.setSurname("Lastcuddles");
114        assertEquals("Lastcuddles", owner1.getSurname());
115    }
116
117    @Test
118    void testSetOwnedCars() {
119        Car car6 = new Car("Ferrari", 990, owner1, 9);
120
121        int carArraySize = owner1.getOwnedCars().length;
122        Car[] ownedCars = new Car[carArraySize];
123        ownedCars[0] = car6;
124
125        // Set the array of ownedCars as the array of cars owned by the Name object
126        owner1.setOwnedCars(ownedCars);
```

```
127         Car[] list = owner1.getOwnedCars();
128         assertEquals("Ferrari", list[0].getModel());
129     }
130     //Methods
131
132     @Test
133     void testGetLastCommaFirst() {
134         assertEquals("Turing, Alan", owner1.getLastCommaFirst());
135     }
136     @Test
137     void testGetFullName_WithMiddleName() {
138         assertEquals("Alan Mathson Turing", owner1.getFullName());
139     }
140     @Test
141     void testGetFullName_WithoutMiddleName() {
142         Name owner2 = new Name();
143         owner2.setFirstName("Patrick");
144         owner2.setSurname("Frendo");
145         assertEquals("Patrick Frendo", owner2.getFullName());
146         owner2 = null;
147     }
148     @Test
149     void testGetFirstAndLast() {
150         assertEquals("Alan Turing", owner1.getFirstAndLast());
151     }
152     //Unit test to determine the 3rd car for a given name object
153     @Test
154     void testGetThirdOwnedCarFromNameObject() {
155         Car[] list = owner1.getOwnedCars();
156         assertEquals("Mitsubishi Colt", list[2].getModel());
157     }
158
159
```