



Software Engineering IT314

Lab-8

Name: Meet Patel

Student ID: 202001074

Date: 19/04/2023

→ Below is the provided code for the 'Boa' class created for testing.

```
// represents a boa constructor  
public class Boa {  
    private String name;  
    private int length; // the length of the boa, in feet  
    private String favoriteFood;  
    public Boa (String name, int length, String favoriteFood){  
        this.name = name;  
        this.length = length;
```

```

    this.favoriteFood = favoriteFood;
}
// returns true if this boa constrictor is healthy
public boolean isHealthy(){
    return this.favoriteFood.equals("granola bars");
}
// returns true if the length of this boa constrictor is
// less than the given cage length
public boolean fitsInCage(int cageLength){
    return this.length < cageLength;
}
}

```

→ Now we create a Junit Test case for testing the above ‘Boa’ class and named it ‘Boa_test’.

→ Below is the Test code for running the **initial test cases**.

```

package test_lab8;
import static org.junit.jupiter.api.Assertions.*;
import org.junit.Before;
import org.junit.jupiter.api.Test;
class Boa_test {
    @Test
    public void testIsHealthyWithFavoriteFoodGranolaBars() {
        Boa boa = new Boa("Benny", 5, "granola bars");
        assertTrue(boa.isHealthy());
    }

    @Test
    public void testIsHealthyWithFavoriteFoodNotGranolaBars() {
        Boa boa = new Boa("Benny", 5, "mice");
        assertFalse(boa.isHealthy());
    }

    @Test

```

```

public void testFitsInCageWhenLengthLessThanCageLength() {
    Boa boa = new Boa("Benny", 5, "granola bars");
    assertTrue(boa.fitsInCage(10));
}

```

@Test

```

public void testFitsInCageWhenLengthGreaterThanCageLength() {
    Boa boa = new Boa("Benny", 20, "granola bars");
    assertFalse(boa.fitsInCage(10));
}
}

```

→ Now we ran the above test cases successfully.

→ Now we use the **@Before** tag to run another test case.

→ The **@Before** tag runs before every test case(**@Test**).

→ The code snippet below show how to execute it.

```

private Boa jen;

```

```

private Boa ken;

```

@Before

```

public void setUp() throws Exception {
    jen = new Boa("Jennifer", 2, "grapes");
    ken = new Boa("Kenneth", 3, "granola bars");
}

```

→ Now performing Q5

→ Below is the test code for the question.

```
@Test
public void testIsHealthy() {
    Boa jen = new Boa("Jen", 5, "granola bars");
    Boa ken = new Boa("Ken", 6, "mice");

    assertTrue(jen.isHealthy());
    assertFalse(ken.isHealthy());
}
```

```
@Test
public void testFitsInCage() {
    Boa jen = new Boa("Jen", 5, "granola bars");
    Boa ken = new Boa("Ken", 6, "mice");

    assertFalse(jen.fitsInCage(2));
    assertTrue(jen.fitsInCage(10));
    assertTrue(jen.fitsInCage(15));
    assertFalse(ken.fitsInCage(4));
    assertTrue(ken.fitsInCage(15));
    assertTrue(ken.fitsInCage(20));
}
```

→ Now creating another method in the 'Boa' class for Q7.

Method Code:

```
public int lengthInInches() {
    return this.length * 12;
}
```

Testing code:

@Test

```
public void testLengthInInches() {  
    Boa boa = new Boa("John", 5, "grapes");  
    int expectedLengthInInches = 60;  
    int actualLengthInInches = boa.lengthInInches();  
    assertEquals(expectedLengthInInches, actualLengthInInches);  
}
```

→ Below is the snippet for the complete testing code.

The screenshot displays the Eclipse IDE interface. The main editor shows the `Boa_test.java` file with the following code:

```
1 package test_lab8;  
2  
3 import static org.junit.jupiter.api.Assertions.*;  
4  
5 import org.junit.Before;  
6 import org.junit.jupiter.api.Test;  
7  
8 class Boa_test {  
9  
10     @Test  
11     public void testIsHealthyWithFavoriteFoodGranolaBars() {  
12         Boa boa = new Boa("Benny", 5, "granola bars");  
13         assertTrue(boa.isHealthy());  
14     }  
15  
16     @Test  
17     public void testIsHealthyWithFavoriteFoodNotGranolaBars() {  
18         Boa boa = new Boa("Benny", 5, "mice");  
19         assertFalse(boa.isHealthy());  
20     }  
21  
22     @Test  
23     public void testFitsInCageWhenLengthLessThanCageLength() {  
24         Boa boa = new Boa("Benny", 5, "granola bars");  
25         assertTrue(boa.fitsInCage(10));  
26     }  
27  
28     @Test  
29     public void testFitsInCageWhenLengthGreaterThanCageLength() {  
30         Boa boa = new Boa("Benny", 20, "granola bars");  
31         assertFalse(boa.fitsInCage(10));  
32     }  
33  
34     private Boa jen;  
35     private Boa ken;  
36  
37     @Before  
38     public void setUp() throws Exception {  
39         jen = new Boa("Jennifer", 2, "grapes");  
40         ken = new Boa("Kenneth", 3, "granola bars");  
41     }  
42 }  
43  
44
```

The left sidebar shows the Project Explorer with the `Boa_test` class selected. The bottom status bar indicates the test run results: `Boa_test [Runner: JUnit 5] (0.014 s)`. The bottom right corner displays the Coverage report:

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Lab_008	50.3 %	74	73	147

Elipse WorkSpace - Lab_008\src\test\java\Boa_test.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer JUnit X

Finished after 0.071 seconds

Runs: 7/7 Errors: 0 Failures: 0

> Boa_test [Runner: JUnit 5] (0.014 s)

```
38 @Before
39 public void setUp() throws Exception {
40     jen = new Boa("Jennifer", 2, "grapes");
41     ken = new Boa("Kenneth", 3, "granola bars");
42 }
43
44
45 @Test
46 public void testLengthInInches() {
47     Boa boa = new Boa("John", 5, "grapes");
48     int expectedLengthInInches = 60;
49     int actualLengthInInches = boa.lengthInInches();
50     assertEquals(expectedLengthInInches, actualLengthInInches);
51 }
52
53 @Test
54 public void testIsHealthy() {
55     Boa jen = new Boa("Jen", 5, "granola bars");
56     Boa ken = new Boa("Ken", 6, "mice");
57     assertTrue(jen.isHealthy());
58     assertFalse(ken.isHealthy());
59 }
60
61
62
63 @Test
64 public void testFitsInCage() {
65     Boa jen = new Boa("Jen", 5, "granola bars");
66     Boa ken = new Boa("Ken", 6, "mice");
67     assertFalse(jen.fitsInCage(2));
68     assertTrue(jen.fitsInCage(10));
69     assertTrue(jen.fitsInCage(15));
70     assertFalse(ken.fitsInCage(4));
71     assertTrue(ken.fitsInCage(15));
72     assertTrue(ken.fitsInCage(20));
73 }
74
75
76
77
78
79 }
80
81
```

Failure Trace

Coverage X

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
> Lab_008	50.3 %	74	73	147

