

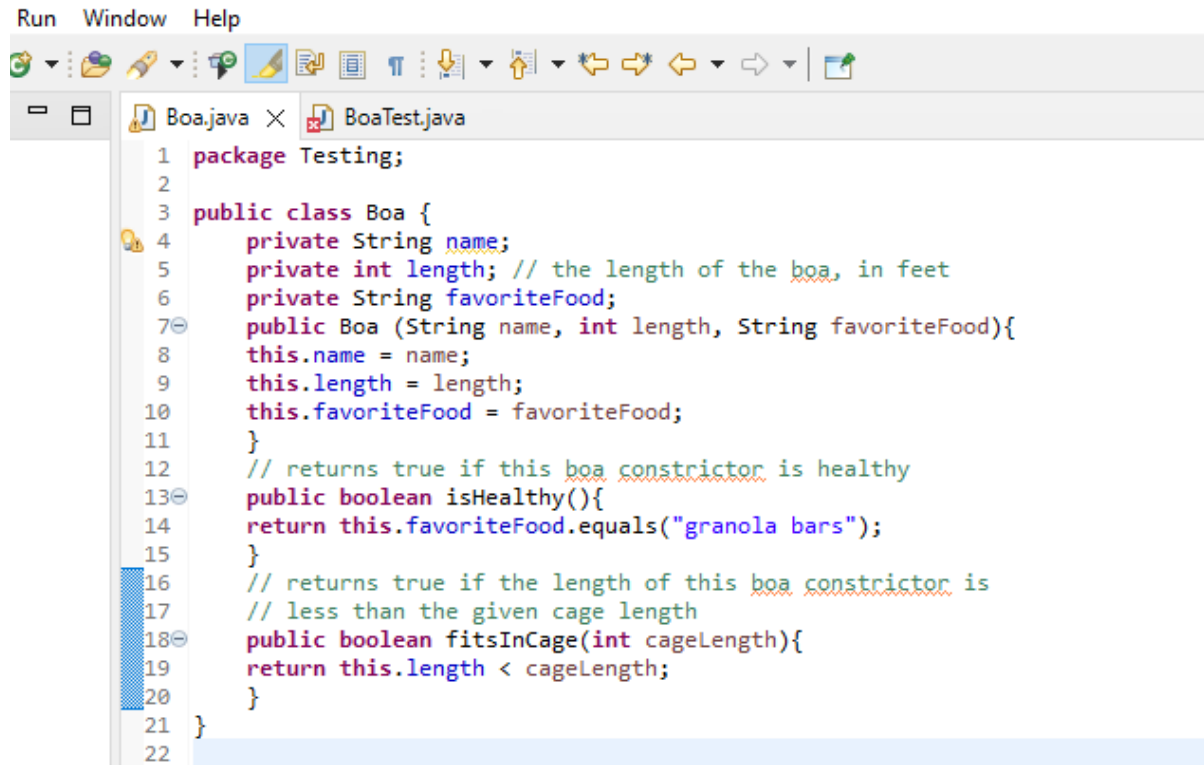
IT314

Software Engineering

Name: Noopur Chaudhary
ID: 202001091

Lab Exercises:

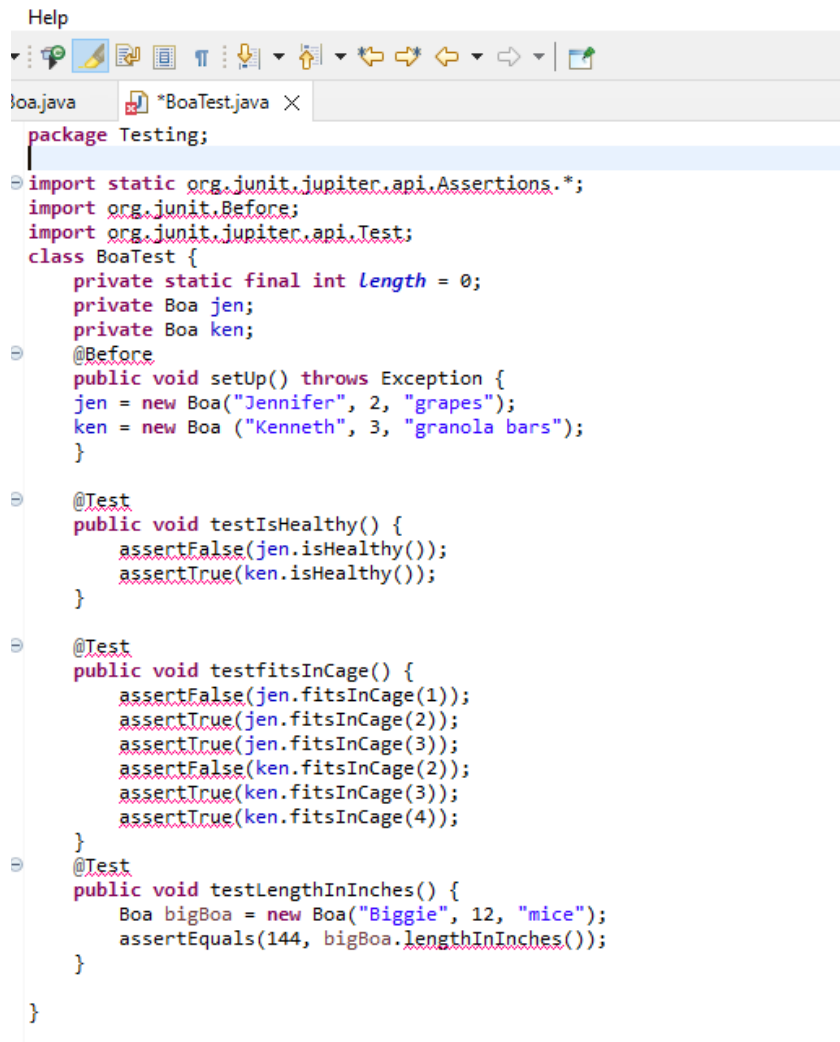
Given class Boa:



The screenshot shows an IDE window with two tabs: 'Boa.java' and 'BoaTest.java'. The 'Boa.java' tab is active, displaying the following Java code:

```
1 package Testing;
2
3 public class Boa {
4     private String name;
5     private int length; // the length of the boa, in feet
6     private String favoriteFood;
7     public Boa (String name, int length, String favoriteFood){
8         this.name = name;
9         this.length = length;
10        this.favoriteFood = favoriteFood;
11    }
12    // returns true if this boa constructor is healthy
13    public boolean isHealthy(){
14        return this.favoriteFood.equals("granola bars");
15    }
16    // returns true if the length of this boa constructor is
17    // less than the given cage length
18    public boolean fitsInCage(int cageLength){
19        return this.length < cageLength;
20    }
21 }
22
```

Test cases:



```
Help
Boa.java *BoaTest.java X
package Testing;

import static org.junit.jupiter.api.Assertions.*;
import org.junit.Before;
import org.junit.jupiter.api.Test;
class BoaTest {
    private static final int length = 0;
    private Boa jen;
    private Boa ken;

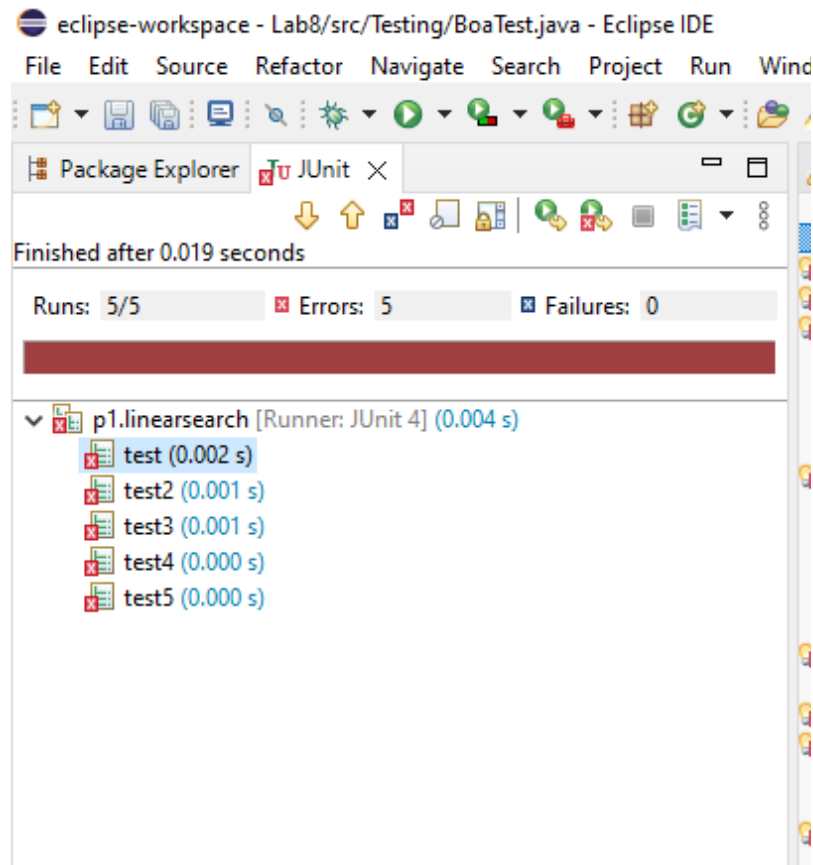
    @Before
    public void setUp() throws Exception {
        jen = new Boa("Jennifer", 2, "grapes");
        ken = new Boa ("Kenneth", 3, "granola bars");
    }

    @Test
    public void testIsHealthy() {
        assertFalse(jen.isHealthy());
        assertTrue(ken.isHealthy());
    }

    @Test
    public void testfitsInCage() {
        assertFalse(jen.fitsInCage(1));
        assertTrue(jen.fitsInCage(2));
        assertTrue(jen.fitsInCage(3));
        assertFalse(ken.fitsInCage(2));
        assertTrue(ken.fitsInCage(3));
        assertTrue(ken.fitsInCage(4));
    }

    @Test
    public void testLengthInInches() {
        Boa bigBoa = new Boa("Biggie", 12, "mice");
        assertEquals(144, bigBoa.lengthInInches());
    }
}
```

Case Testing:



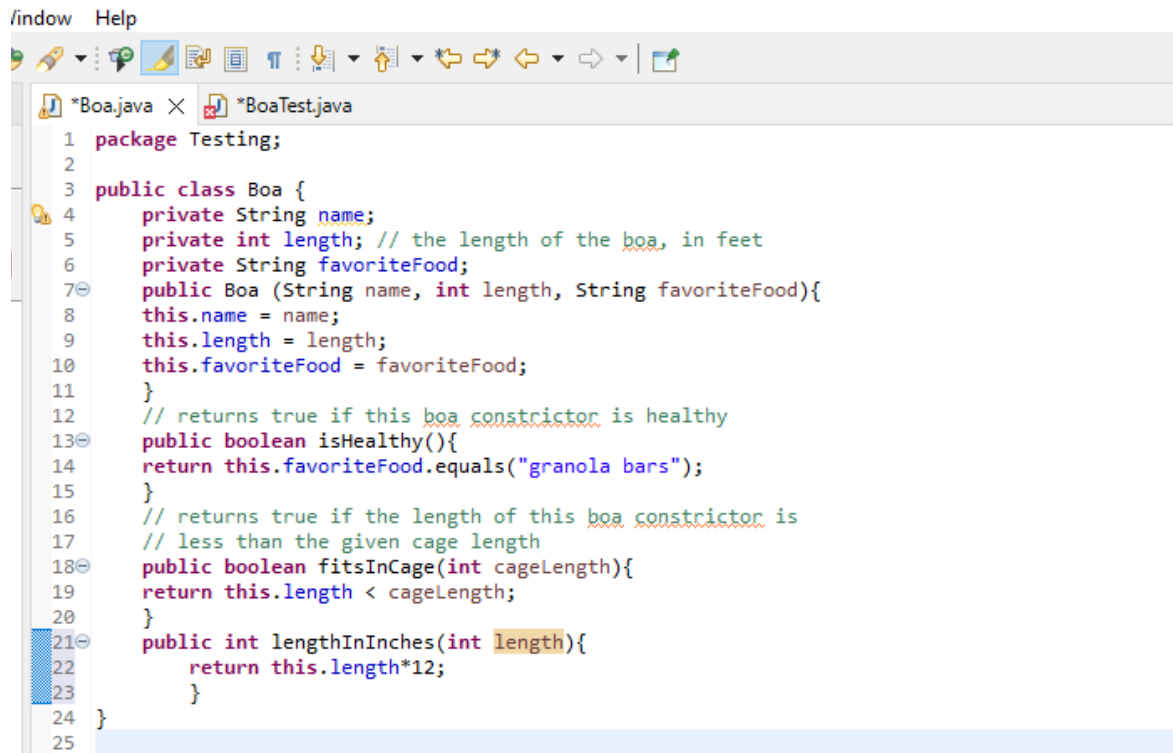
Test Cases for FitsInCage()

```
w Help
Boa.java *BoaTest.java X
33     Boa bigBoa = new Boa("Biggie", 12, "mice");
34     assertEquals(144, bigBoa.lengthInInches());
35 }
36
37 @Test
38 public void testFitsInCage1() {
39     int cage = 4;
40     boolean output = jen.fitsInCage(cage);
41     assertEquals(output, false);
42 }
43 @Test
44 public void testFitsInCage2() {
45     int cage = 6;
46     boolean output = ken.fitsInCage(cage);
47     assertEquals(output, false);
48 }
49 @Test
50 public void testFitsInCage3() {
51     int cage = 11;
52     boolean output = ken.fitsInCage(cage);
53     assertEquals(output, true);
54 }
55 @Test
56 public void testFitsInCage4() {
57     int cage = 2;
58     boolean output = ken.fitsInCage(cage);
59     assertEquals(output, false);
60 }
61 }
```

Test Cases for IsHealthy()

```
2
3 @Test
4 public void testIsHealthy_1() {
5     boolean output = ken.isHealthy();
6     assertEquals(output, true);
7 }
8 @Test
9 public void testIsHealthy_2() {
10     boolean output = jen.isHealthy();
11     assertEquals(output, false);
12 }
13
14
15 }
```

New method to the Boa class:



```
1 package Testing;
2
3 public class Boa {
4     private String name;
5     private int length; // the length of the boa, in feet
6     private String favoriteFood;
7     public Boa (String name, int length, String favoriteFood){
8         this.name = name;
9         this.length = length;
10        this.favoriteFood = favoriteFood;
11    }
12    // returns true if this boa constructor is healthy
13    public boolean isHealthy(){
14        return this.favoriteFood.equals("granola bars");
15    }
16    // returns true if the length of this boa constructor is
17    // less than the given cage length
18    public boolean fitsInCage(int cageLength){
19        return this.length < cageLength;
20    }
21    public int lengthInInches(int length){
22        return this.length*12;
23    }
24 }
25
```