Name - Saurav

Kapadiya

Sid - 202001033

# Lab-8

1]Boa.java

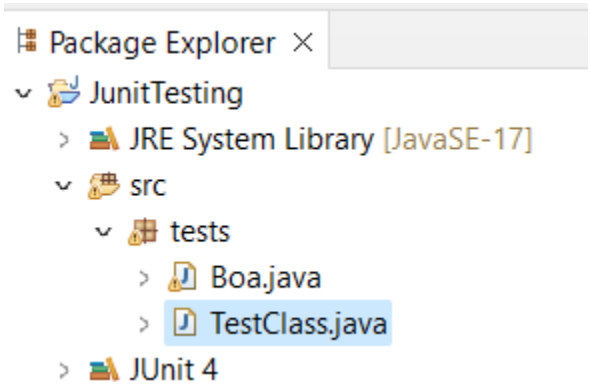
Code:

```
publicclassBoa{
        privateStringname;
        privateintlength;/thelengthoftheboa,infeetpr
        ivateStringfavoriteFood;
        publicBoa(Stringname,intlength,StringfavoriteFood){this.na
                me=name;
                this.length =
                length;this.favoriteFood=favoriteF
                ood;
        }
        /returnstrueifthisboaconstrictorishealthypu
```

```
blicbooleanisHealthy(){
        returnthis.favoriteFood.equals("granolabars");
}
```

```
        /returnstrueifthelengthofthisboaconstrictoris
        /lessthanthegivencagelength
        publicbooleanfitsInCage(intcageLength){ret
                urnthis.length<cageLength;
        }
}
```

# 1}CREATING A NEW ECLIPSEPROJECT AND APACKAGEWITHINTHEPROJECT




# 2}CREATINGACLASSFORBOA

Boa.java × TestClass.java

```java
package tests;

// represents a boa constrictor
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;
    }
}
```

# 3}CREATING A JUNIT TEST CASE WITHMETHODSISHEALTHYANDFITSINCAGE

Boa.java | TestClass.java

```java
package tests;

import static org.junit.Assert.*;
import org.junit.Before;
import org.junit.Test;

public class TestClass {
    private Boa jen, ken;

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

    @Test
    public void isHealthy() {
        fail("Not yet implemented");
    }

    @Test
    public void fitsInCage() {

    }
}
```

4}creating setup method and annotating with@beforeandcreatingjenandkenobjectsofboaclass

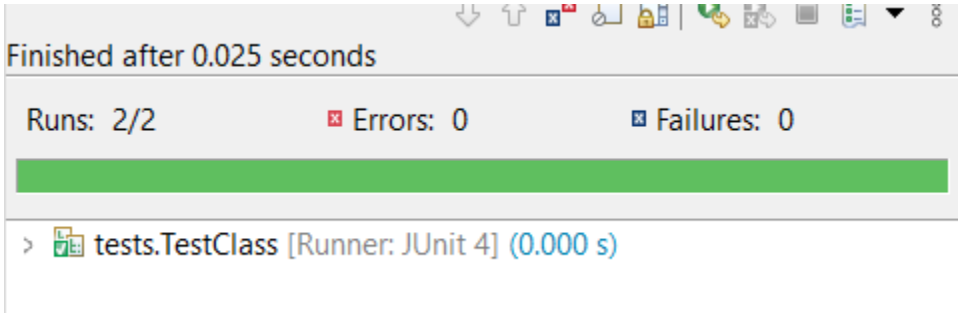
```
public class TestClass {
    private Boa jen, ken;

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

## 5}writingtestsforfitsincageandishealthymethod

```
16      @Test
17      public void testIsHealthy() {
18          assertFalse(jen.isHealthy());
19          assertTrue(ken.isHealthy());
20      }
21
22      @Test
23      public void testFitsInCage() {
24          assertTrue(jen.fitsInCage(3));
25          assertFalse(jen.fitsInCage(2));
26          assertFalse(jen.fitsInCage(1));
27          assertFalse(jen.fitsInCage(0));
28          assertFalse(jen.fitsInCage(-1));
29          assertTrue(ken.fitsInCage(10));
30          assertFalse(ken.fitsInCage(3));
31          assertFalse(ken.fitsInCage(0));
32          assertFalse(ken.fitsInCage(-1));
33      }
34  }
```

## 6}runningbothtests




```
24      // produces the length of the Boa in inches
25      public int lengthInInches(){
26          return 12*this.length;
27
28      }
29  }
```

## writingthetestsforlengthininches

```
35
36      @Test
37      public void testLengthInInches() {
38          assertEquals(24, jen.lengthInInches());
39          assertEquals(36, ken.lengthInInches());
40      }
41  }
```

## runningthetests

FinishedafterD.D22seconds

Runs:3/3 Errors:D Failures:D

› tests.TestClass [Runner: JUnit 4] (0.000 s)