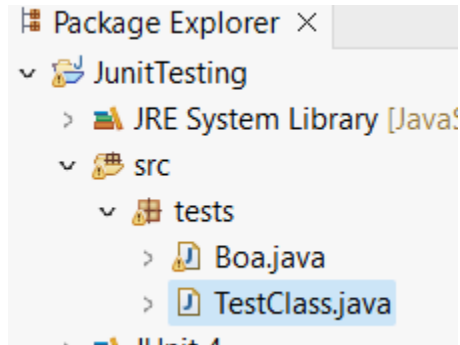


ID: 202001020

Name: Aditya Dineshbhai Patel

Lab 8

1) creating a new Eclipse project and a package inside it.



2) Creating class for Boa

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

3) making a junit test case that fitsInCage and isHealthy.

```

Boa.java TestClass.java ×
1 package tests;
2
3 import static org.junit.Assert.*;
4 import org.junit.Before;
5 import org.junit.Test;
6
7 public class TestClass {
8     private Boa jen, ken;
9
10    @Before
11    public void setUp() throws Exception {
12        jen = new Boa("Jennifer", 2, "grapes");
13        ken = new Boa("Kenneth", 3, "granola bars");
14    }
15
16    @Test
17    public void isHealthy() {
18        fail("Not yet implemented");
19    }
20
21    @Test
22    public void fitsInCage() {
23

```

4) Creating setup method and annotating with @Before and creating jen and ken objects of Boa class.

```

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) Writing tests for fitsInCage and isHealthy method

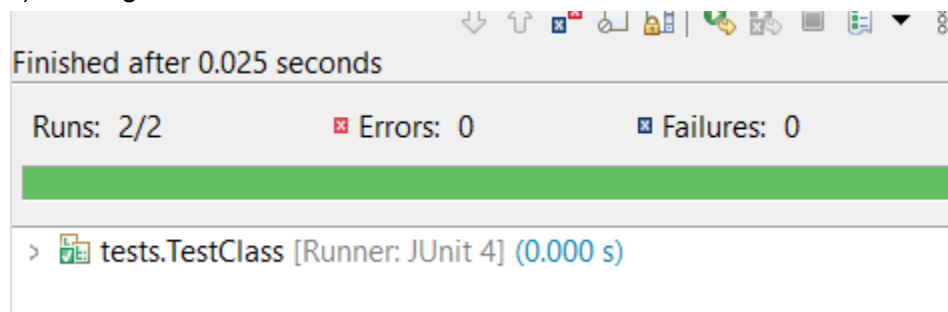
```

5 @Test
7 public void testIsHealthy() {
8     assertFalse(jen.isHealthy());
9     assertTrue(ken.isHealthy());
10 }

11 @Test
12 public void testFitsInCage() {
13     assertTrue(jen.fitsInCage(3));
14     assertFalse(jen.fitsInCage(2));
15     assertFalse(jen.fitsInCage(1));
16     assertFalse(jen.fitsInCage(0));
17     assertFalse(jen.fitsInCage(-1));
18     assertTrue(ken.fitsInCage(10));
19     assertFalse(ken.fitsInCage(3));
20     assertFalse(ken.fitsInCage(0));
21     assertFalse(ken.fitsInCage(-1));
22 }

```

6) running both tests



7) Writing code to convert length in inches

```

// produces the length of the
public int lengthInInches() {
    return 12*this.length;
}

```

Writing the tests for length in inches.


```
public void testLengthInInches() {  
    assertEquals(24, jen.lengthInInches());  
    assertEquals(36, ken.lengthInInches());  
}
```

Running the tests

Runs: 3/3

Errors: 0

Failures: 0

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