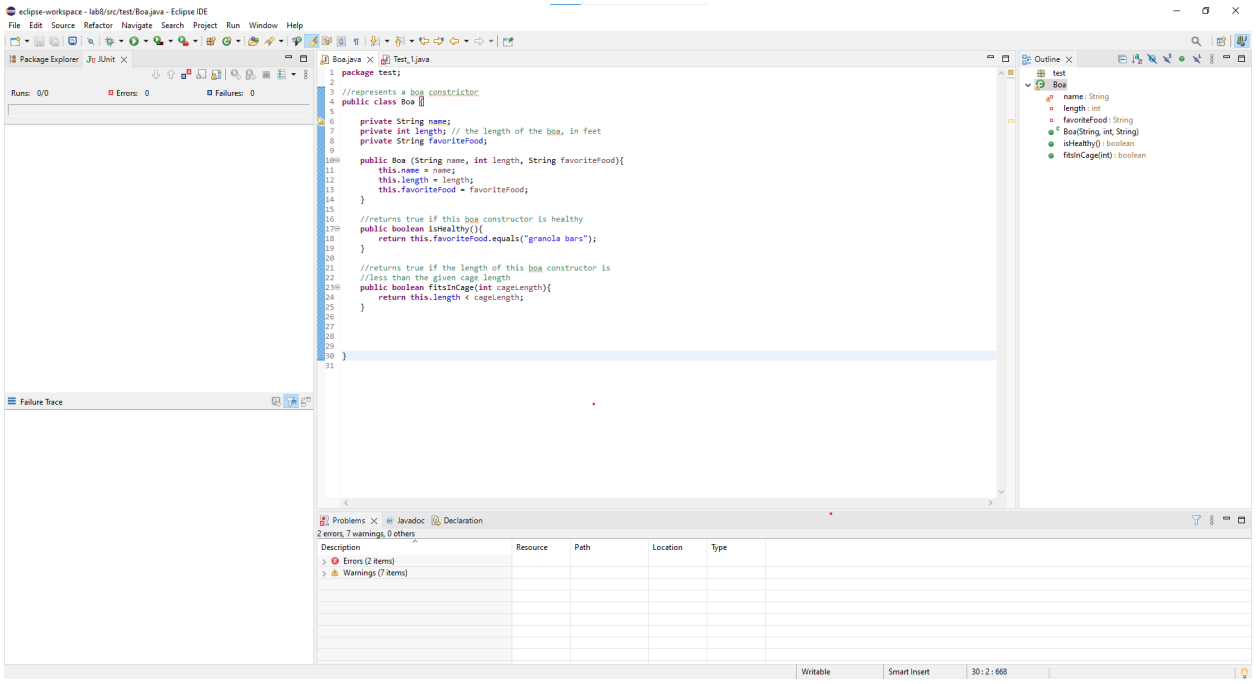
Name : Anurag Ganvit
ID : 202001187

Lab-8 : Unit Testing with JUnit

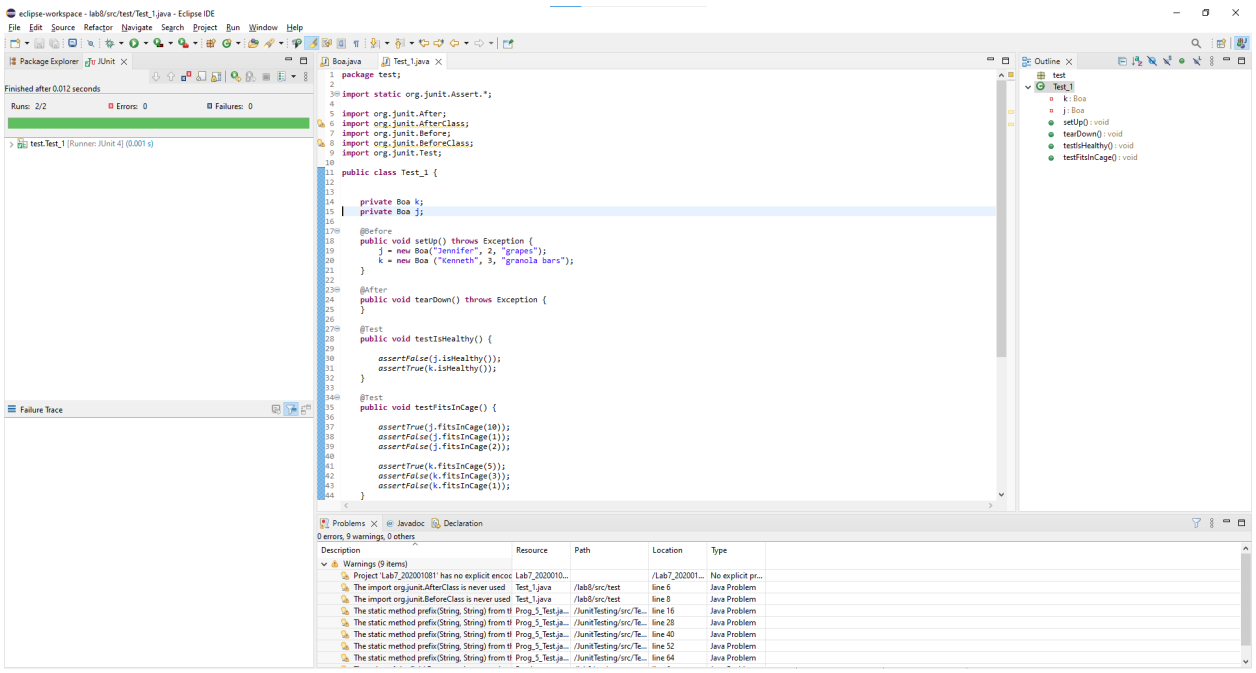
1. Create a new Eclipse project, and within the project create a package.



2. Test method to test the behavior of the Boa class :

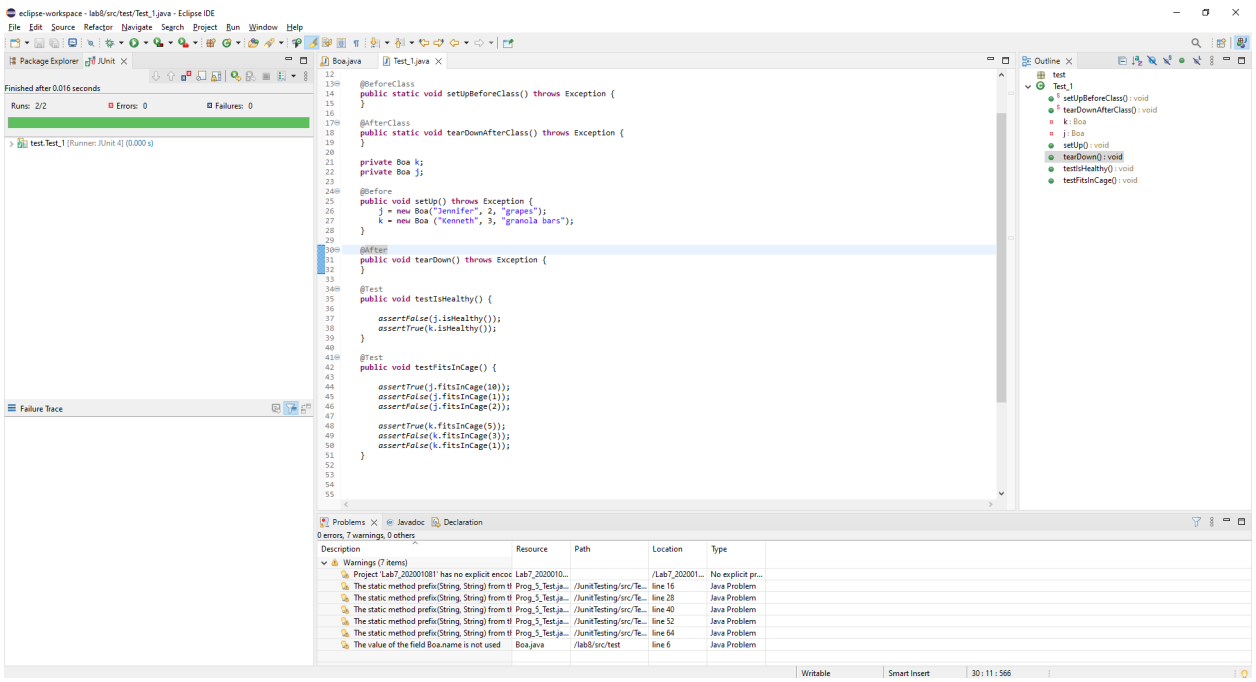
```
import org.junit.After;
import org.junit.AfterClass;
```

```
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;

        @Test
        public void testIsHealthy() {

    assertFalse(j.isHealthy());
    assertTrue(k.isHealthy());
        }
        @Test
        public void testFitsInCage() {
    assertTrue(j.fitsInCage(10));
    assertFalse(j.fitsInCage(1));
    assertFalse(j.fitsInCage(2));

    assertTrue(k.fitsInCage(5));
    assertFalse(k.fitsInCage(3));
    assertFalse(k.fitsInCage(1));
        }
}
```



3. Modified setUp() method in the BoaTest class :

```
public static void tearDownAfterClass() throws Exception {
	}
	private Boa k;
	private Boa j;
	@Before
	public void setUp() throws Exception {
		j = new Boa("Jennifer", 2, "grapes");
		k = new Boa ("Kenneth", 3, "granola bars");
	}
```



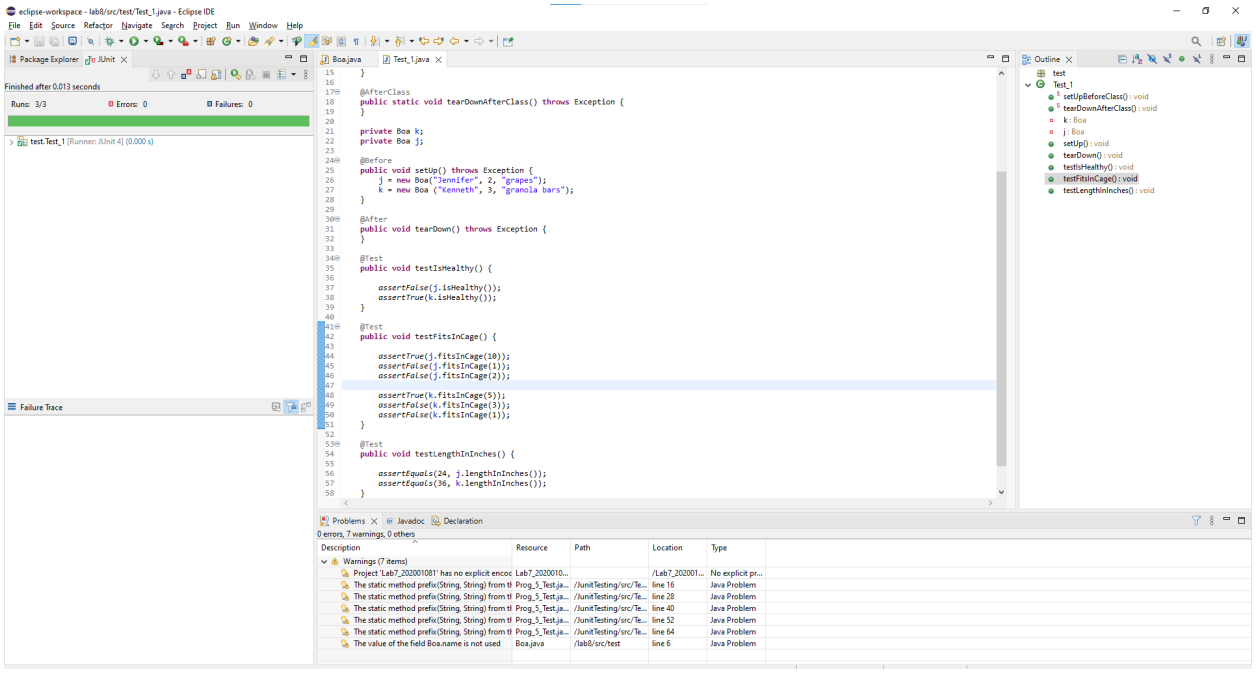
4. Modified testIsHealthy() method in the BoaTest class :

```
@Test
public void testIsHealthy() {
    // check that jen is not healthy
    assertFalse(jen.isHealthy());

    // check that ken is healthy
    assertTrue(ken.isHealthy());
}
```

5. Modified testFitsInCage() method in the BoaTest class :

```
@Test
        public void testFitsInCage() {
    assertTrue(j.fitsInCage(10));
    assertFalse(j.fitsInCage(1));
    assertFalse(j.fitsInCage(2));
    assertTrue(k.fitsInCage(5));
    assertFalse(k.fitsInCage(3));
    assertFalse(k.fitsInCage(1));
        }
```

6. Running test cases



7. Here's the modified Boa class with the new lengthInInches() method:

```
package test;
//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 constructor is healthy
	public boolean isHealthy(){
		return this.favoriteFood.equals("granola bars");
	}

	//returns true if the length of this boa constructor is
	//less than the given cage length
	public boolean fitsInCage(int cageLength){
		return this.length < cageLength;
	}


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

8. Add a new test case to the BoaTest class that tests the lengthInInches() method.

```
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
public class Test_1 {
	@BeforeClass
	public static void setUpBeforeClass() throws Exception {
	}
	@AfterClass
	public static void tearDownAfterClass() throws Exception {
	}
	private Boa k;
	private Boa j;


@Test
```

```
    public void testFitsInCage() {
assertTrue(j.fitsInCage(10));
assertFalse(j.fitsInCage(1));
assertFalse(j.fitsInCage(2));
assertTrue(k.fitsInCage(5));
assertFalse(k.fitsInCage(3));
assertFalse(k.fitsInCage(1));
    }
```

This new test case checks that the lengthInInches() method returns the expected value when called on each of the Boa objects created in the setUp() method. It uses the assertEquals() method to compare the expected value to the actual value returned by the lengthInInches() method. The @Test annotation indicates that this is a test method that should be run by JUnit.