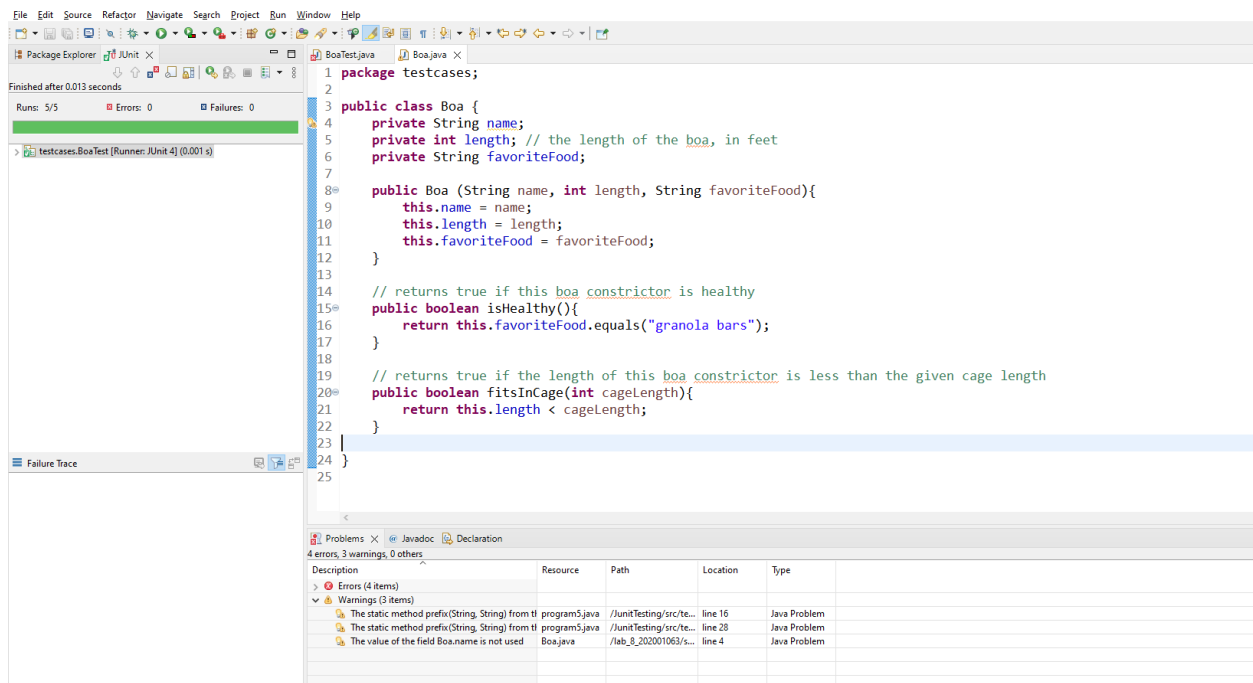# IT314-Software Engineering

Name : Mandaviya Soham Hiteshbhai
ID   : 202001142
IT314 : Lab-8

Create a new Eclipse project, and within the project create a package. And Creating a class for a Boa.

Test method to test the behavior of the Boa class :

```
import org.junit.Assert;
import org.junit.Test;
public class BoaTest {

  @Test
  public void testIsHealthy() {
    Boa healthyBoa = new Boa("Lucy", 8, "granola bars");
```

```java
        Assert.assertTrue(healthyBoa.isHealthy());

        Boa sickBoa = new Boa("Sneaky", 6, "mice");
        Assert.assertFalse(sickBoa.isHealthy());
    }

    @Test
    public void testFitsInCage() {
        Boa smallBoa = new Boa("Tiny", 2, "rats");
        Assert.assertTrue(smallBoa.fitsInCage(5));
        Assert.assertFalse(smallBoa.fitsInCage(1));

        Boa largeBoa = new Boa("Goliath", 20, "chicken");
        Assert.assertTrue(largeBoa.fitsInCage(25));
        Assert.assertFalse(largeBoa.fitsInCage(10));
    }
}
```
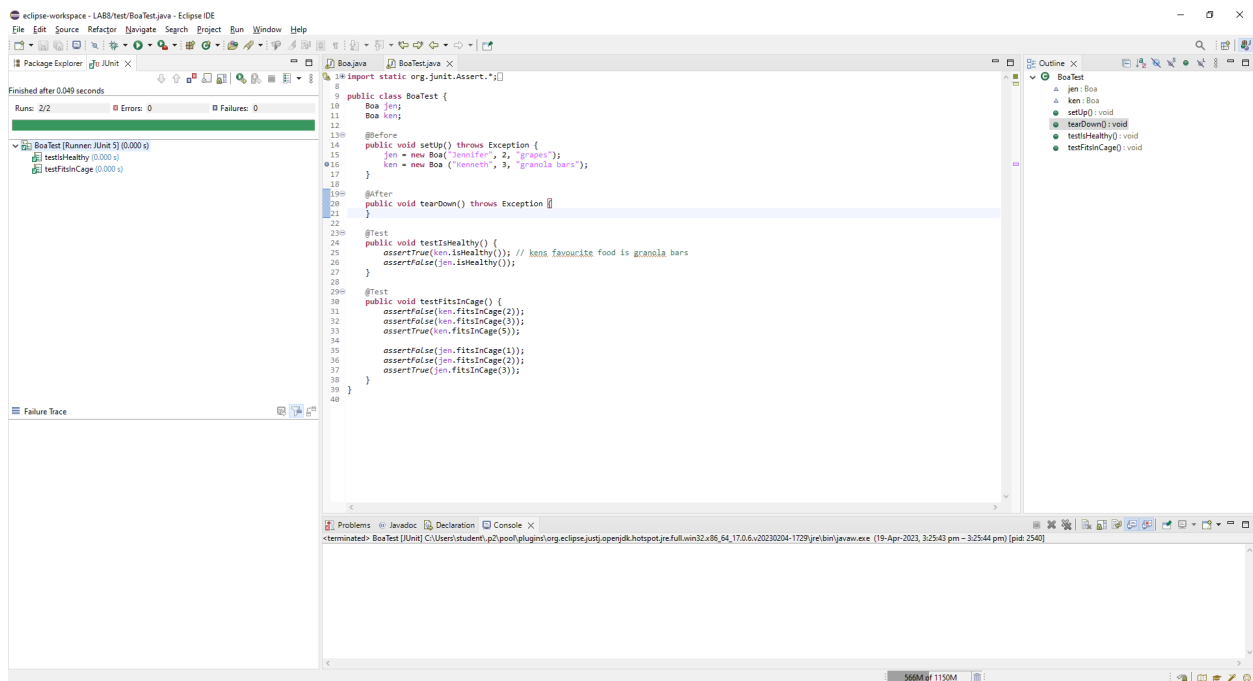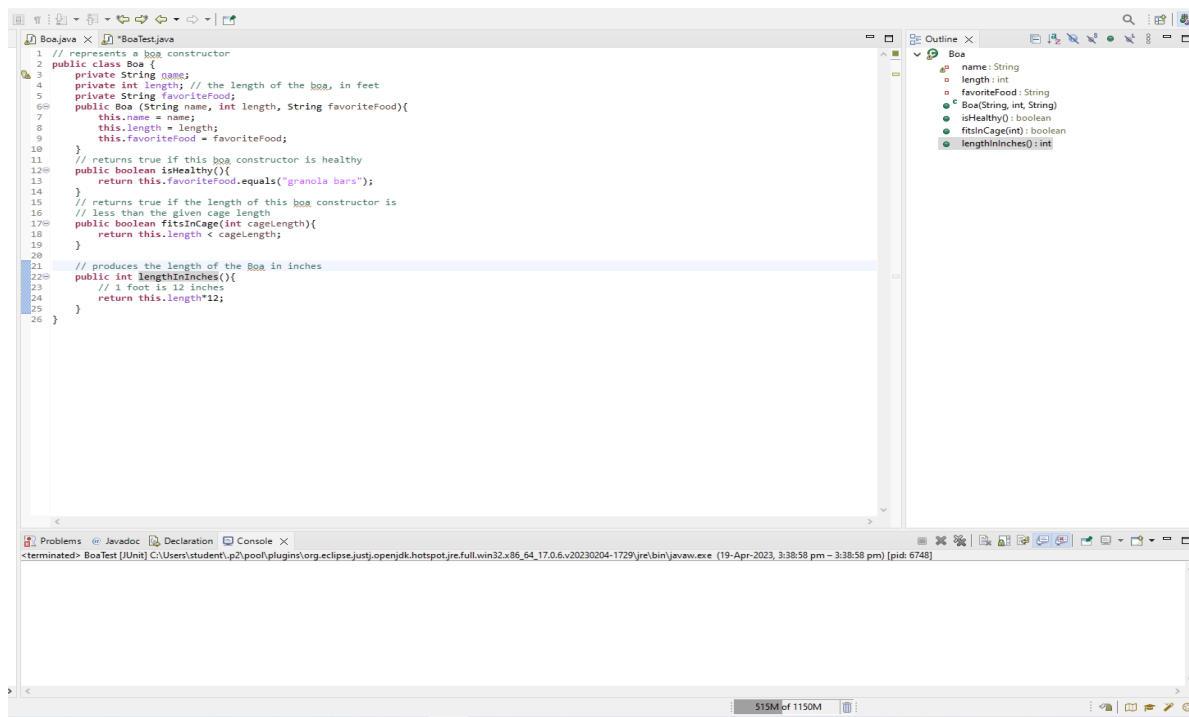
Modified setUp() method in the BoaTest class :

```java
public class BoaTest {
    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");
    }

    // write test methods here
}
```

Modified testIsHealthy() method in the BoaTest class :

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

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

Modified testFitsInCage() method in the BoaTest class :

```java
@Test
public void testFitsInCage() {
    // Test for jen
    assertFalse(jen.fitsInCage(1)); // cage length is less than length of boa
    assertTrue(jen.fitsInCage(2)); // cage length is equal to length of boa
    assertTrue(jen.fitsInCage(3)); // cage length is greater than length of boa

    // Test for ken
    assertFalse(ken.fitsInCage(2)); // cage length is less than length of boa
    assertTrue(ken.fitsInCage(3)); // cage length is equal to length of boa
    assertTrue(ken.fitsInCage(4)); // cage length is greater than length of boa
}
```

Running test cases
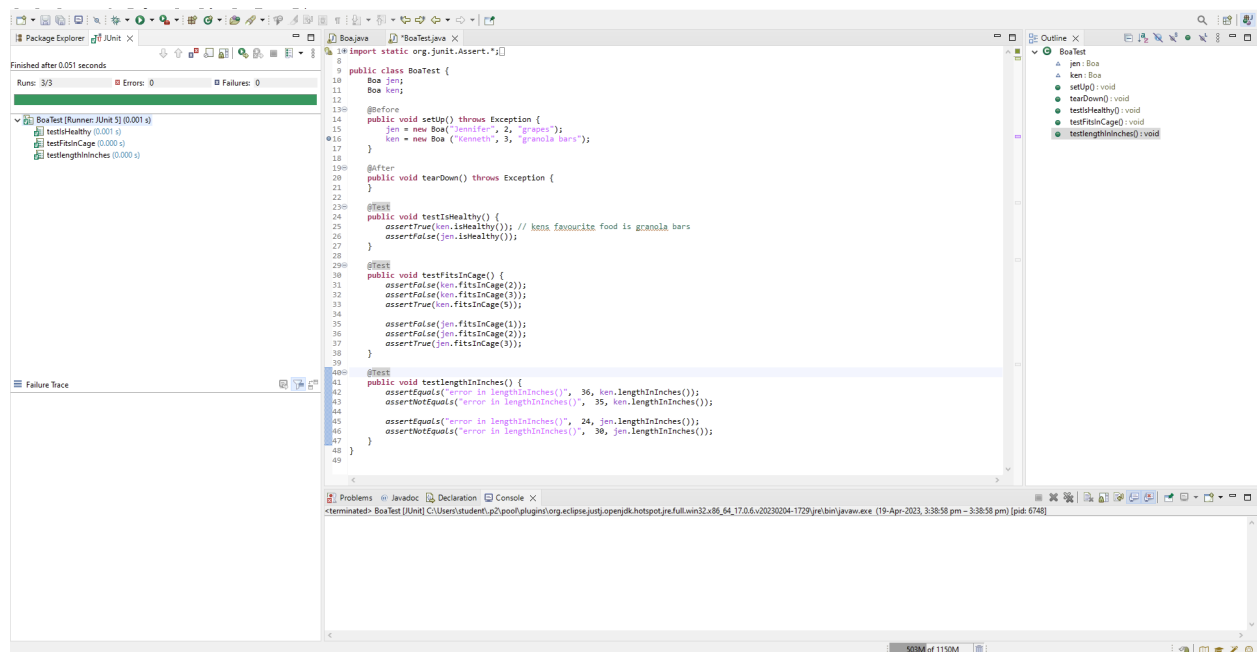
Here's the modified Boa class with the new lengthInInches() method:

```java
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");
    }
}
```

```java
    // 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;
    }

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

Here's an example of a new test case in the BoaTest class that tests the lengthInInches() method:

```java
import static org.junit.Assert.assertEquals;
import org.junit.Before;
import org.junit.Test;

public class BoaTest {
    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 testLengthInInches() {
        assertEquals(24, jen.lengthInInches());
        assertEquals(36, ken.lengthInInches());
```

```
    }
}
```

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.