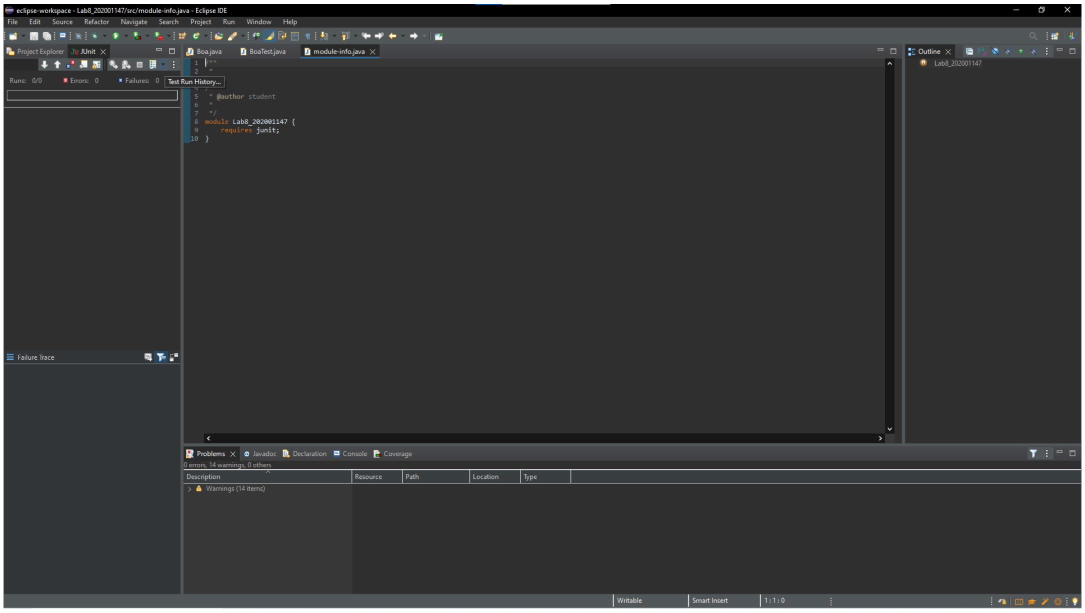
IT314 - Software Engineering

Lab 8 : Unit testing with Junit
Name : Parth Kantibhai Patel
Id : 202001147

1. First I created a new Java project in eclipse with the name Lab8_20200147 and created a package inside it with the name mypackage.



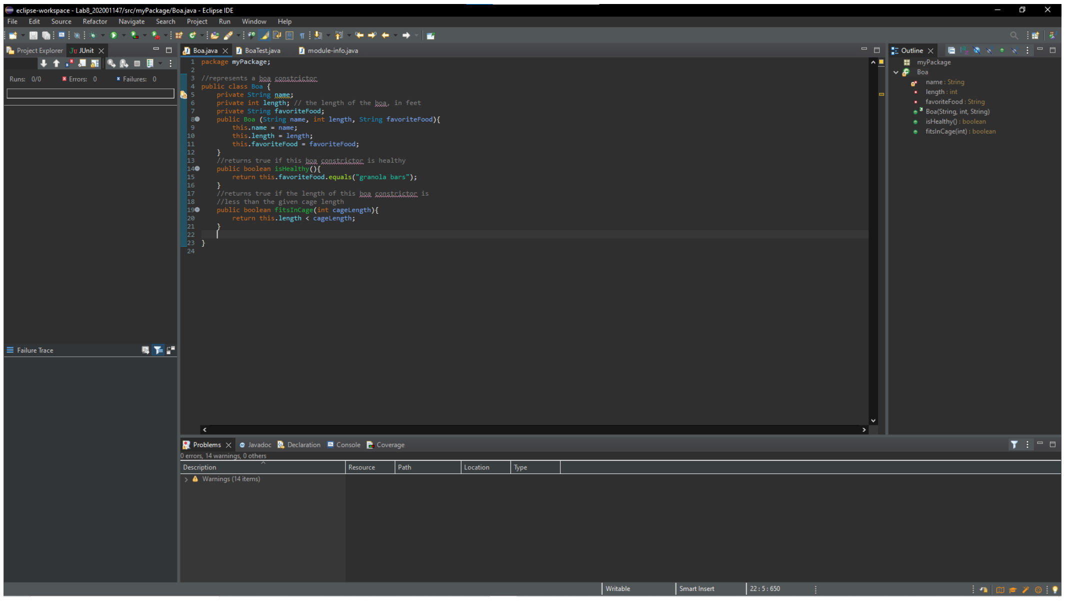
eclipse-workspace - Lab8_202001147/src/module-info.java - Eclipse IDE

```
/**
 *
 * @author student
 *
 */
module Lab8_202001147 {
    requires junit;
}
```

0 errors, 14 warnings, 0 others

Warnings (14 items)

2. Then as mentioned i created a class named Boa and added the given code of manual inside it.



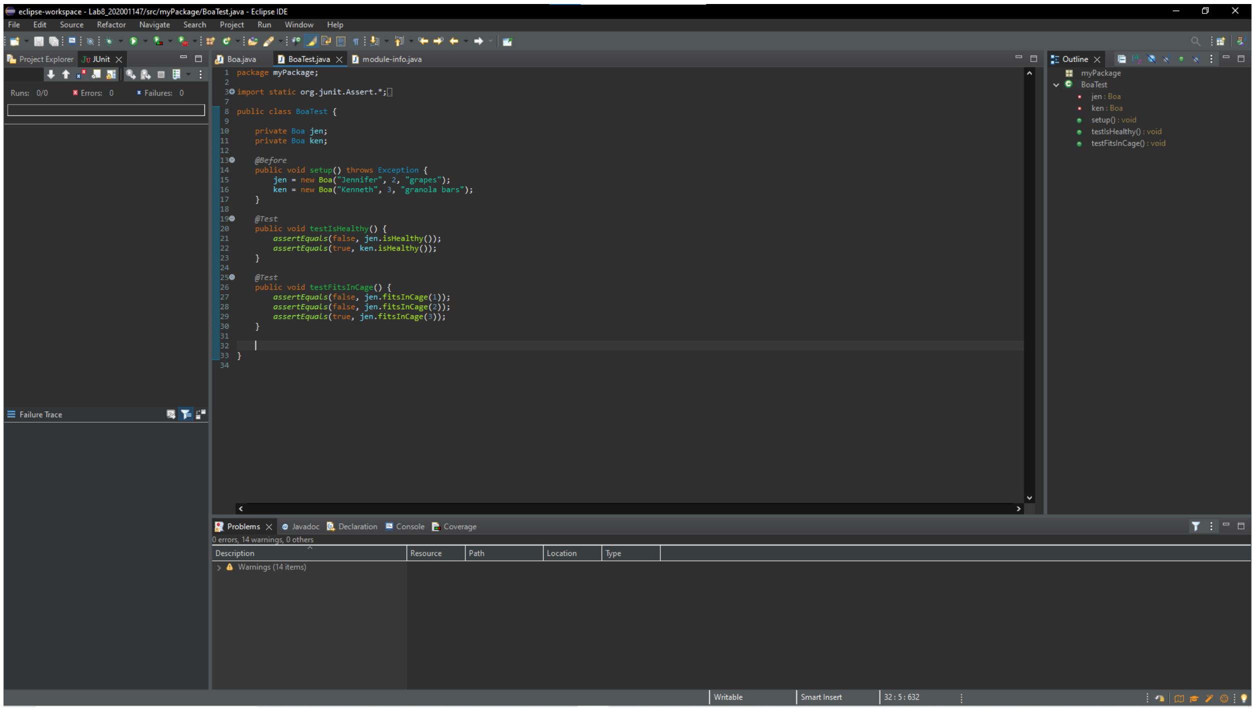
eclipse-workspace - Lab8_202001147/src/myPackage/Boa.java - Eclipse IDE

```
package myPackage;

//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;
    }

}
```

0 errors, 14 warnings, 0 others

Warnings (14 items)

3. After that I created a JUnit test file for the Boa Class with the name BoaTest.



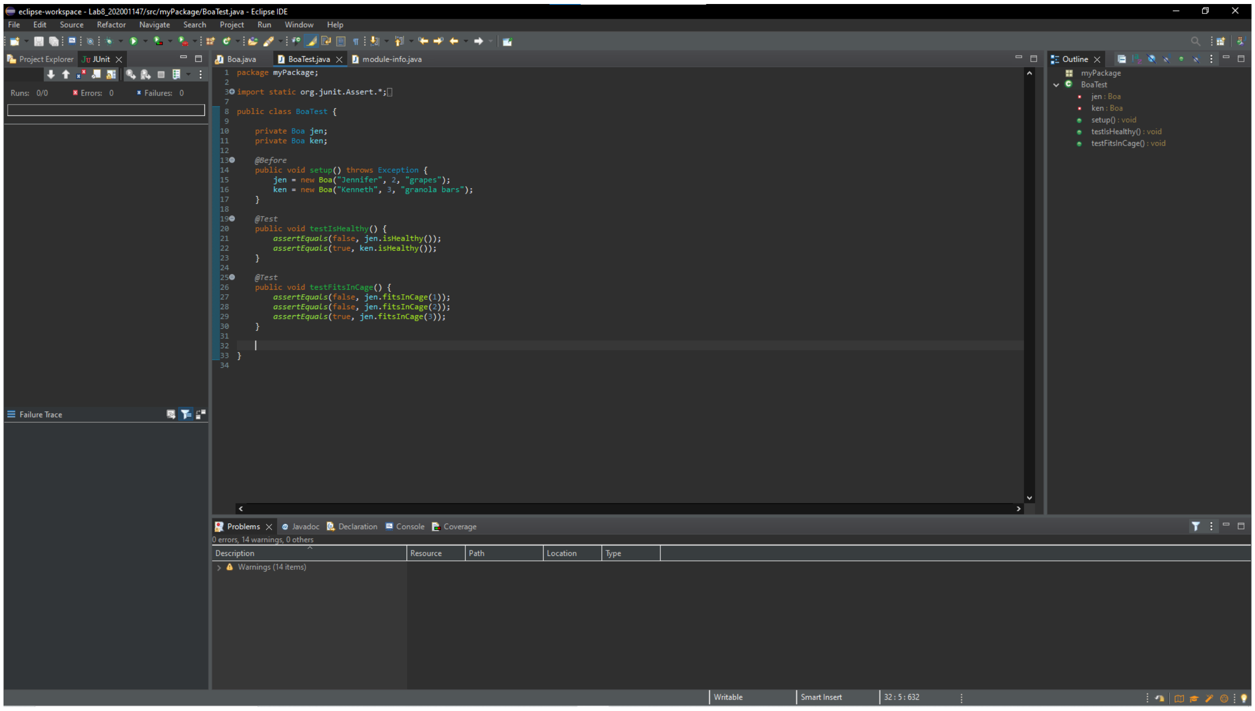
```java
package myPackage;

import static org.junit.Assert.*;

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 testIsHealthy() {
        assertEquals(false, jen.isHealthy());
        assertEquals(true, ken.isHealthy());
    }

    @Test
    public void testFitsInCage() {
        assertEquals(false, jen.fitsInCage(1));
        assertEquals(false, jen.fitsInCage(2));
        assertEquals(true, jen.fitsInCage(3));
    }

}
```

0 errors, 14 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (14 items)				

4. Then I modified the setUp method to initialize the variables.

5. After that I implemented tests for the given two functions testIsHealthy() vans testFitsInCage().



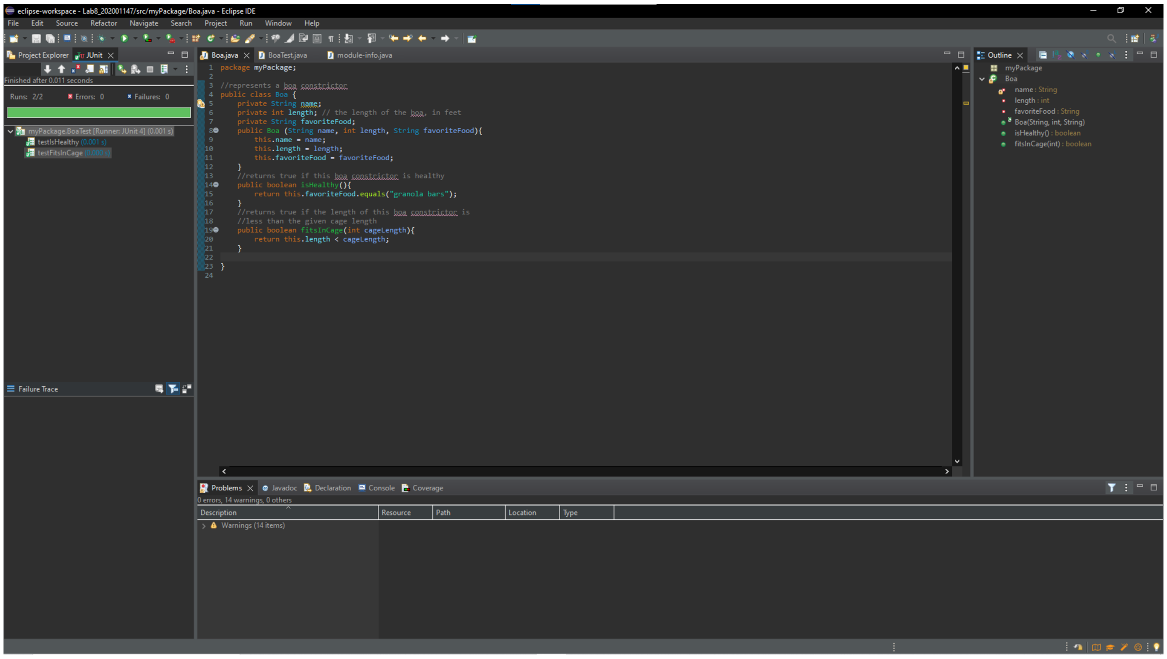
```java
package myPackage;

import static org.junit.Assert.*;

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 testIsHealthy() {
        assertEquals(false, jen.isHealthy());
        assertEquals(true, ken.isHealthy());
    }

    @Test
    public void testFitsInCage() {
        assertEquals(false, jen.fitsInCage(1));
        assertEquals(false, jen.fitsInCage(2));
        assertEquals(true, jen.fitsInCage(3));
    }

}
```

0 errors, 14 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (14 items)				

For testing the fitsInCage() function, there is no need to write tests for both jen and ken objects as the function is same for both jen and ken objects and the output of test cases only depends where=her the given length is greater than, less than or equal to actual length of object. The behavior will be similar in both cases.

6. I ran the JUnit test file and all the tests passed. There are no red bars in the output. It can be seen that test cases have been passed 2 out of 2.



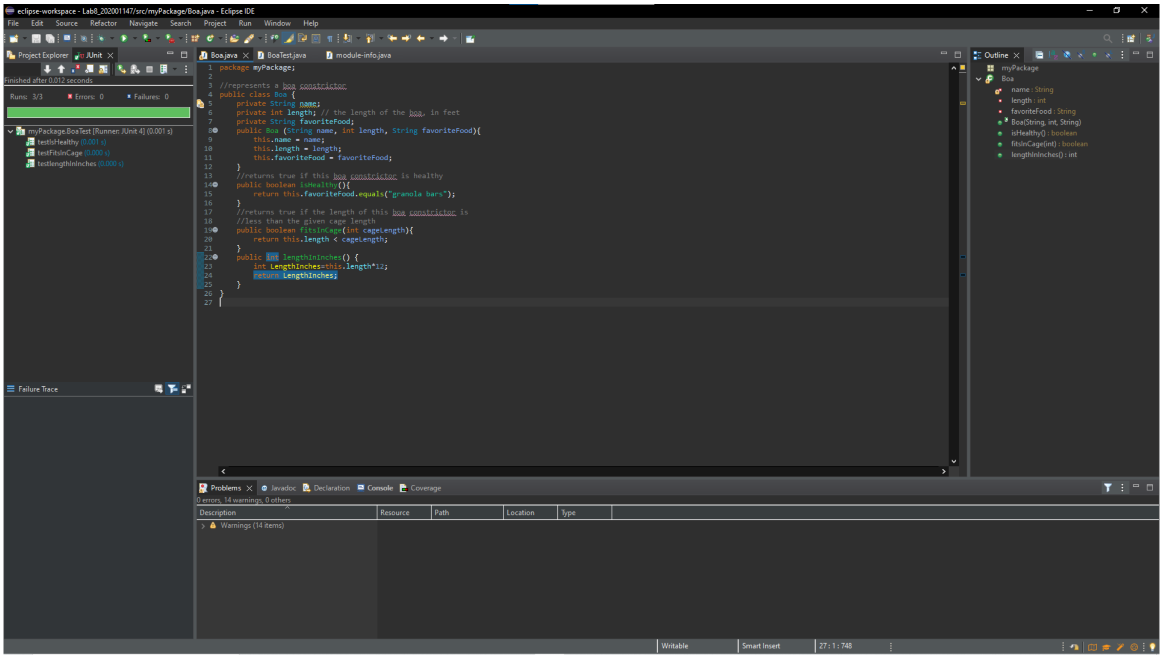
eclipse-workspace - Lab8_202001147/src/myPackage/Boa.java - Eclipse IDE

```
Finished after 0.011 seconds
Runs: 2/2    Errors: 0    Failures: 0
myPackage.BoaTest [Runner: JUnit 4] (0.001 s)
    testIsHealthy (0.001 s)
    testFitsInCage (0.000 s)
```

```java
package myPackage;

//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;
    }

}
```

```
0 errors, 14 warnings, 0 others
Warnings (14 items)
```

7. I added a new method to the Boa class with the name lengthInInches() to get the length in inches.



eclipse-workspace - Lab8_202001147/src/myPackage/Boa.java - Eclipse IDE

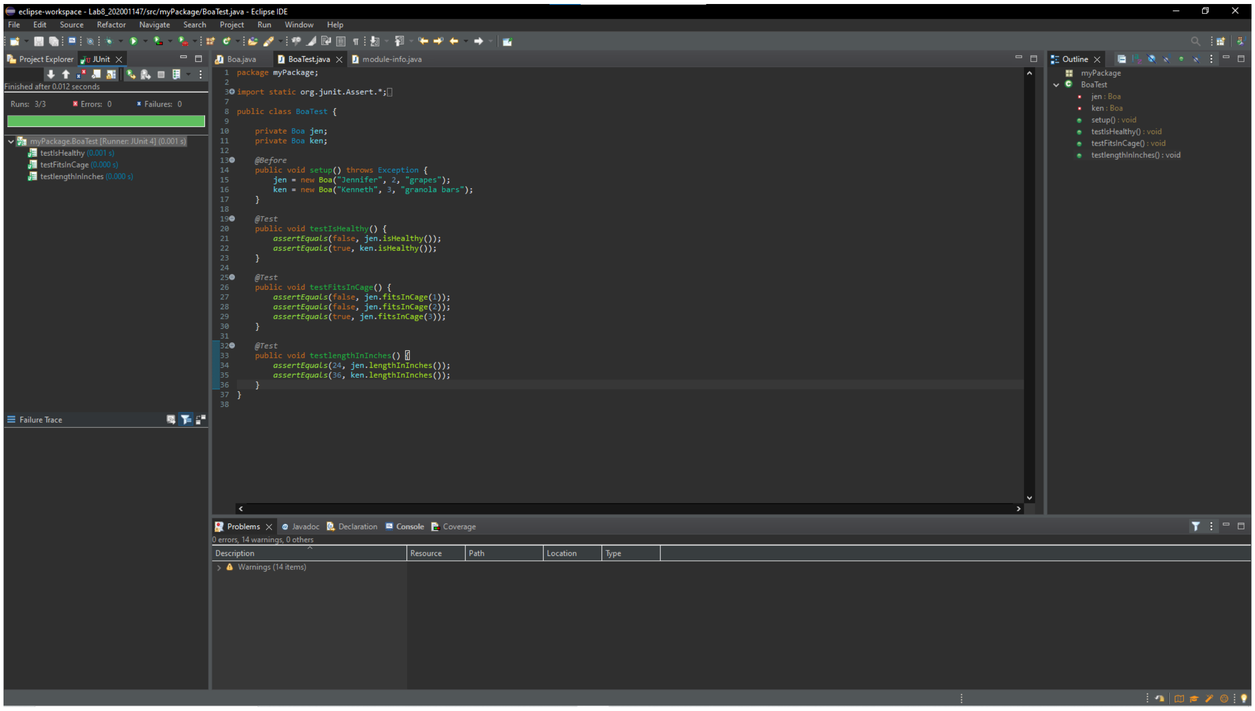
```
Finished after 0.012 seconds
Runs: 3/3    Errors: 0    Failures: 0
myPackage.BoaTest [Runner: JUnit 4] (0.001 s)
    testIsHealthy (0.001 s)
    testFitsInCage (0.000 s)
    testlengthInInches (0.000 s)
```

```java
package myPackage;

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

```
0 errors, 14 warnings, 0 others
Warnings (14 items)
```

As the length of the Boa is given in feet, to convert it into inches, I multiplied the length with 12 and returned the value.

8. Then I wrote another test case for this new method and ran the 3 test cases together.



eclipse-workspace - Lab8_202001147/src/myPackage/BoaTest.java - Eclipse IDE

Finished after 0.012 seconds

Runs: 3/3 Errors: 0 Failures: 0

```
myPackage.BoaTest [Runner: JUnit 4] (0.001 s)
    testIsHealthy (0.001 s)
    testFitsInCage (0.000 s)
    testlengthInInches (0.000 s)
```

```java
package myPackage;

import static org.junit.Assert.*;

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 testIsHealthy() {
        assertEquals(false, jen.isHealthy());
        assertEquals(true, ken.isHealthy());
    }

    @Test
    public void testFitsInCage() {
        assertEquals(false, jen.fitsInCage(1));
        assertEquals(false, jen.fitsInCage(2));
        assertEquals(true, jen.fitsInCage(3));
    }

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

0 errors, 14 warnings, 0 others

Warnings (14 items)

Thus, test cases have been written for the given Boa class and all the three methods have been tested with required Junit test cases.