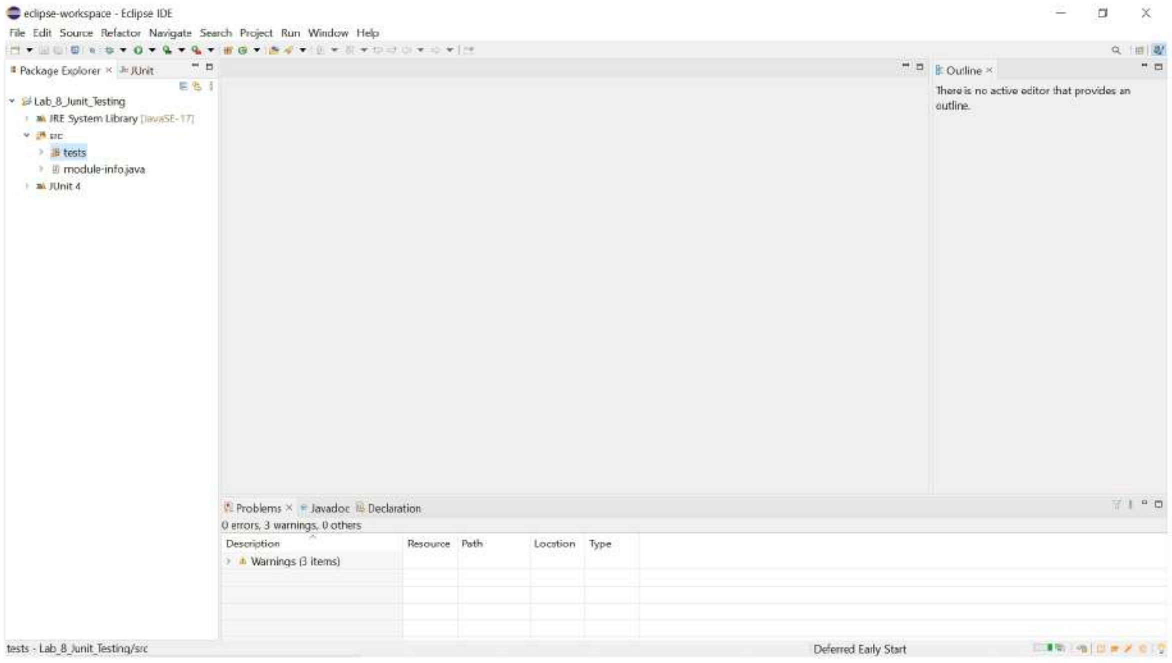
IT314: Software Engineering

Lab Report

DATE	21/04/2023
STUDENT ID	202001416
TITLE	Junit
SUBMISSION NO.	8

Create a new Eclipse project,


eclipse-workspace - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
Outline
Problems
Javadoc
Declaration

Lab_8_Junit_Testing
JRE System Library [JavaSE-17]
src
tests
module-info.java
JUnit 4

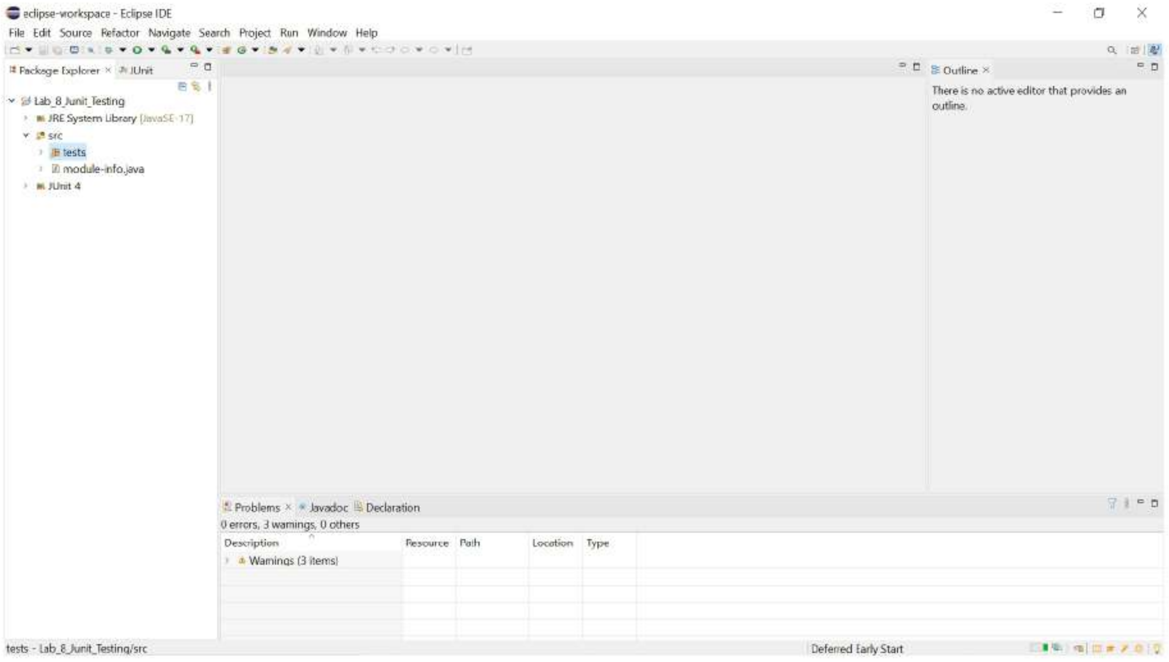
There is no active editor that provides an outline.

0 errors, 3 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (3 items)				

tests - Lab_8_Junit_Testing/src
Deferred Early Start

Create a new class named "Boa" with the following code,


eclipse-workspace - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
Outline
Problems
Javadoc
Declaration

Lab_8_Junit_Testing
JRE System Library [JavaSE-17]
src
tests
module-info.java
JUnit 4

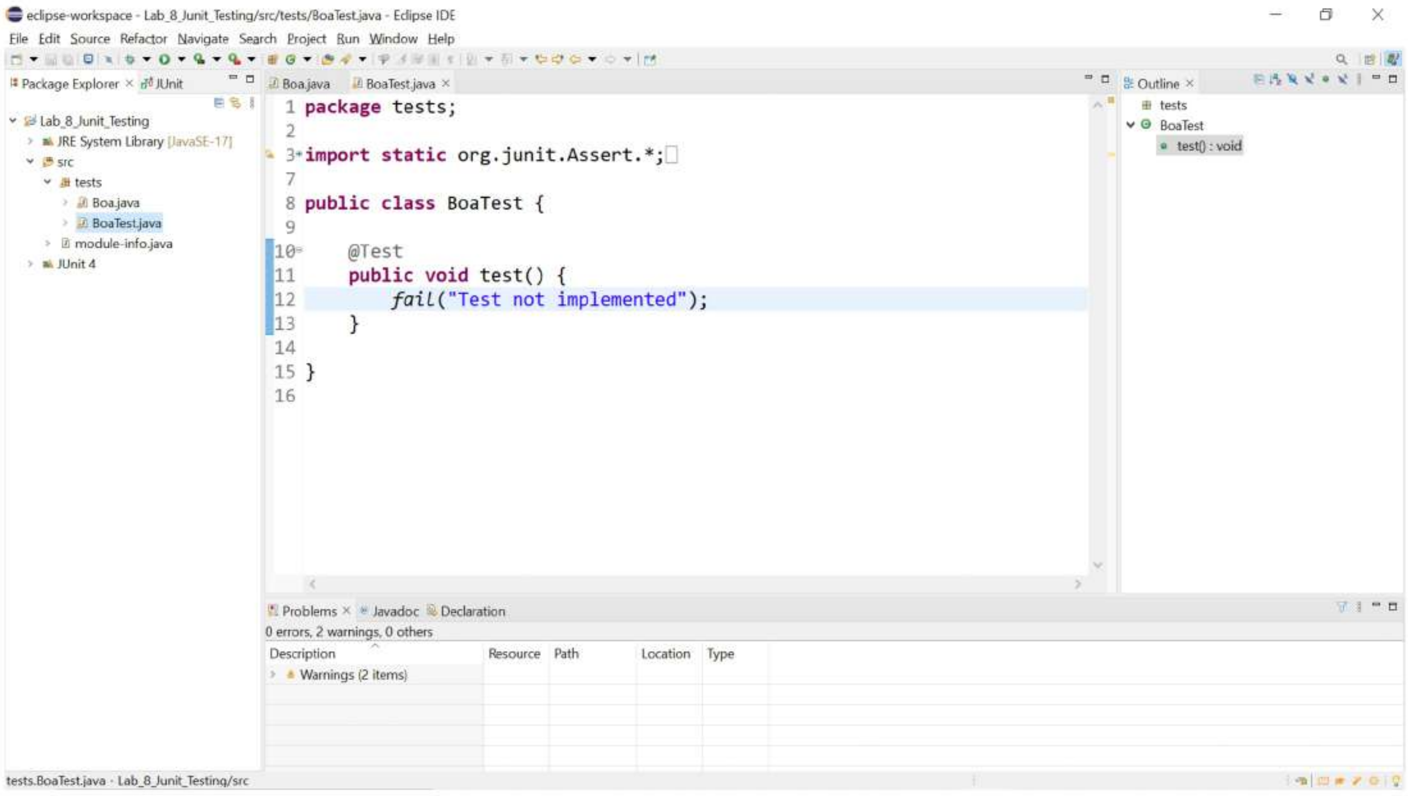
There is no active editor that provides an outline.

0 errors, 3 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (3 items)				

tests - Lab_8_Junit_Testing/src
Deferred Early Start

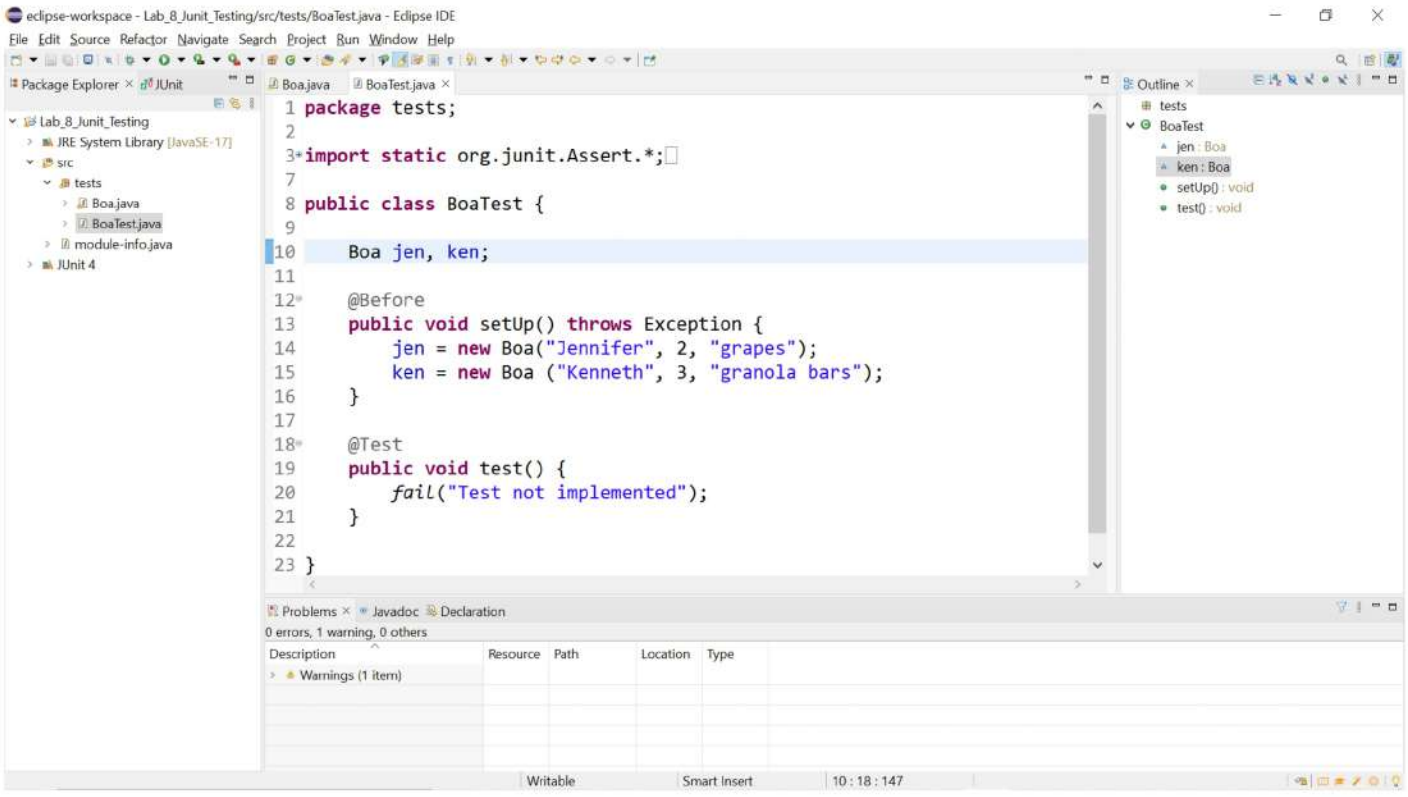
Create a JUnit test case file named "BoaTest",


eclipse-workspace - Lab_8_Junit_Testing/src/tests/BoaTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
Lab_8_Junit_Testing
JRE System Library [JavaSE-17]
src
tests
Boa.java
BoaTest.java
module-info.java
JUnit 4
Outline
tests
BoaTest
test() : void
Problems
Javadoc
Declaration
tests.BoaTest.java - Lab_8_Junit_Testing/src

```java
package tests;

import static org.junit.Assert.*;

public class BoaTest {

    @Test
    public void test() {
        fail("Test not implemented");
    }

}
```

```
0 errors, 2 warnings, 0 others
Description          Resource   Path   Location   Type
> Warnings (2 items)
```

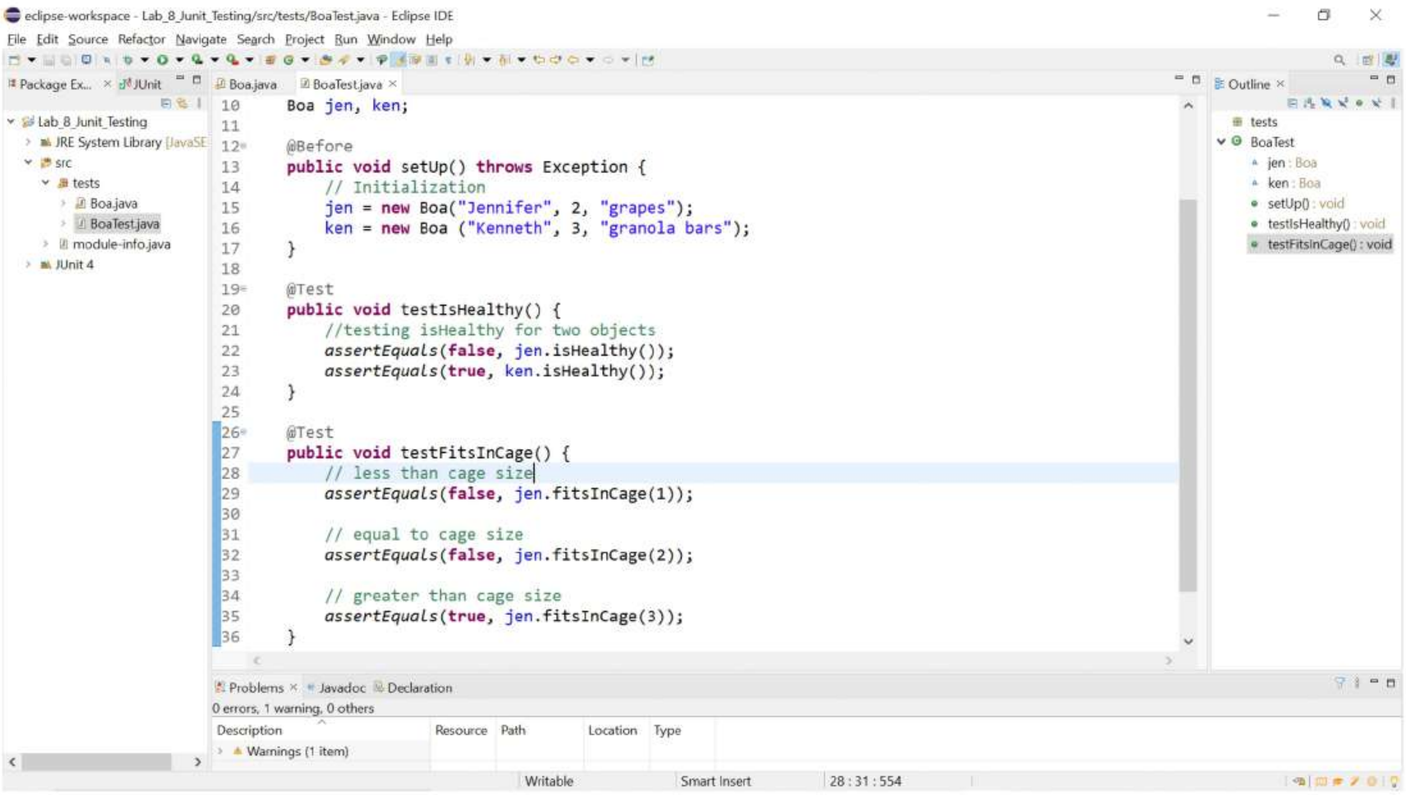
Create a setup function which initializes two new instances of Bos class named "jen" and "ken",



```java
package tests;

import static org.junit.Assert.*;

public class BoaTest {

    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 test() {
        fail("Test not implemented");
    }

}
```

```
0 errors, 1 warning, 0 others
Description          Resource   Path   Location   Type
> Warnings (1 item)
```

Now we have to implement the testIsHealthy() and testFitsInCage() functions in the "BoaTest" class.

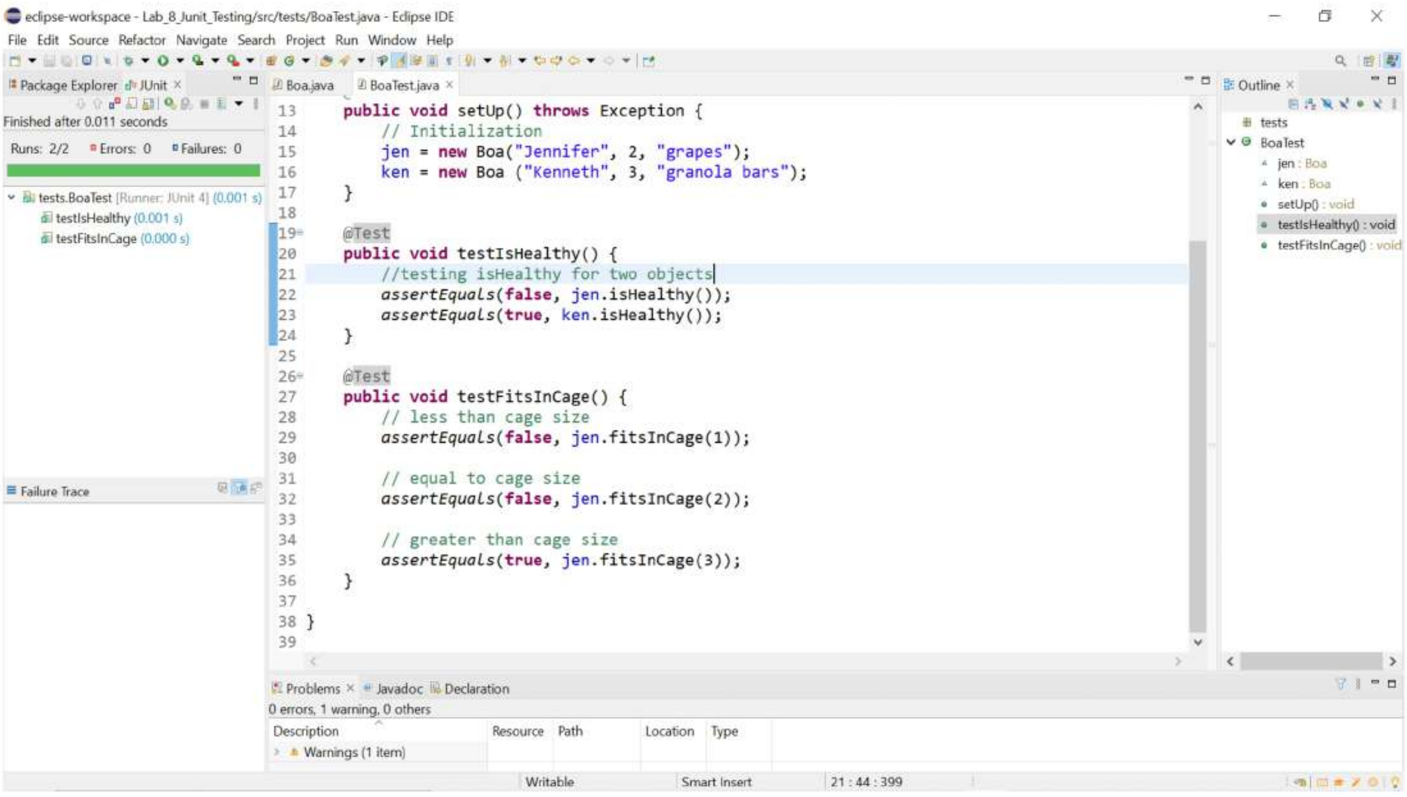

eclipse-workspace - Lab_8_Junit_Testing/src/tests/BoaTest.java - Eclipse IDE

```
    Boa jen, ken;

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

    @Test
    public void testIsHealthy() {
        //testing isHealthy for two objects
        assertEquals(false, jen.isHealthy());
        assertEquals(true, ken.isHealthy());
    }

    @Test
    public void testFitsInCage() {
        // less than cage size
        assertEquals(false, jen.fitsInCage(1));

        // equal to cage size
        assertEquals(false, jen.fitsInCage(2));

        // greater than cage size
        assertEquals(true, jen.fitsInCage(3));
    }
```

```
0 errors, 1 warning, 0 others
Description          Resource   Path   Location   Type
> Warnings (1 item)
```

It is not necessary to develop tests for both ken and jen objects in order to test the fitsInCage() method because the function is the same for both, and the results of test cases depend only on whether the specified length is greater than, less than, or equal to the actual length of the object. In both situations, the behavior will be comparable.

Running the testcases,

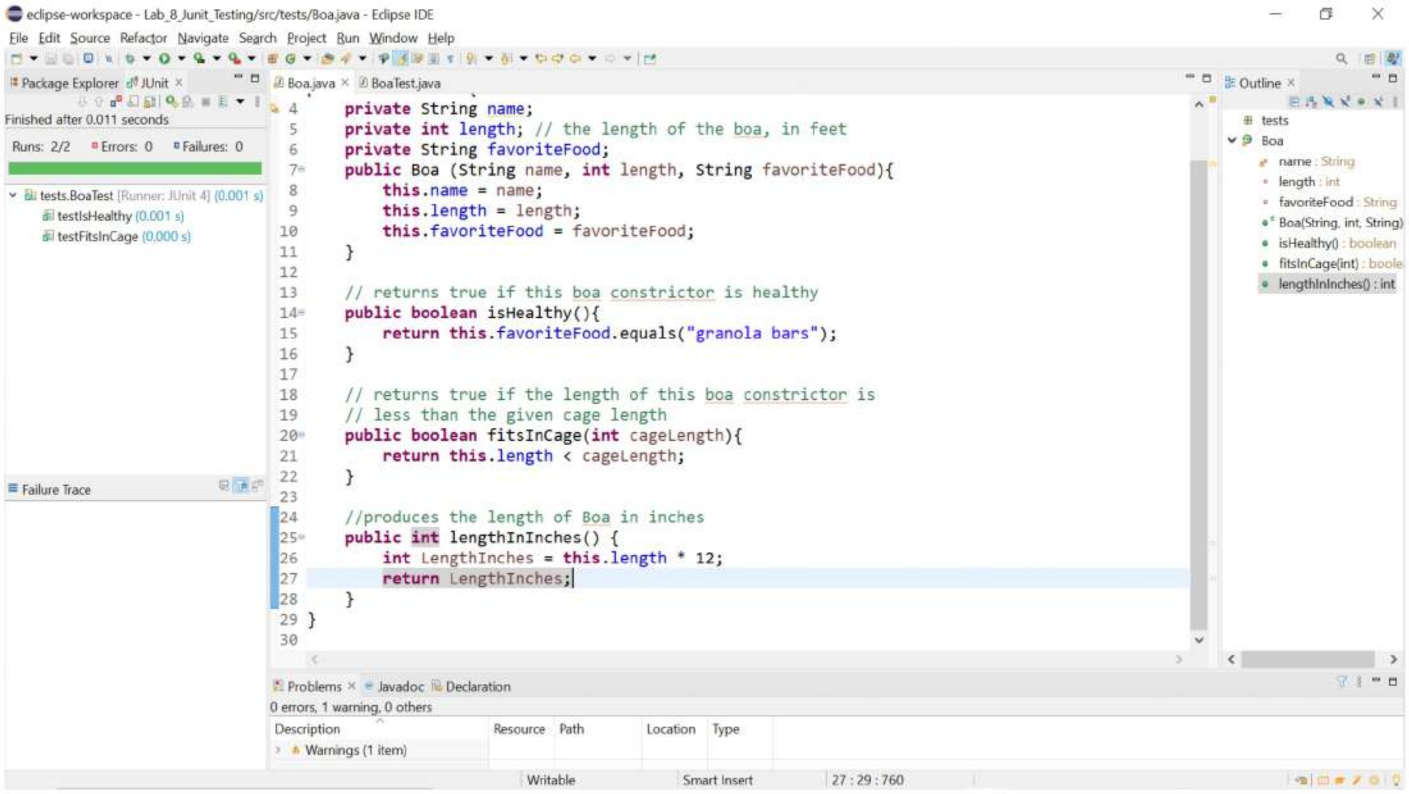

eclipse-workspace - Lab_8_Junit_Testing/src/tests/BoaTest.java - Eclipse IDE

```
Finished after 0.011 seconds
Runs: 2/2     Errors: 0     Failures: 0

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

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

    @Test
    public void testIsHealthy() {
        //testing isHealthy for two objects
        assertEquals(false, jen.isHealthy());
        assertEquals(true, ken.isHealthy());
    }

    @Test
    public void testFitsInCage() {
        // less than cage size
        assertEquals(false, jen.fitsInCage(1));

        // equal to cage size
        assertEquals(false, jen.fitsInCage(2));

        // greater than cage size
        assertEquals(true, jen.fitsInCage(3));
    }

}

```

```
0 errors, 1 warning, 0 others
Description          Resource   Path   Location   Type
> Warnings (1 item)
```

We can observe that both the tests ran successfully.
Now, we create a new method to the Boa class with name lenghtInInches() to get the length in inches.



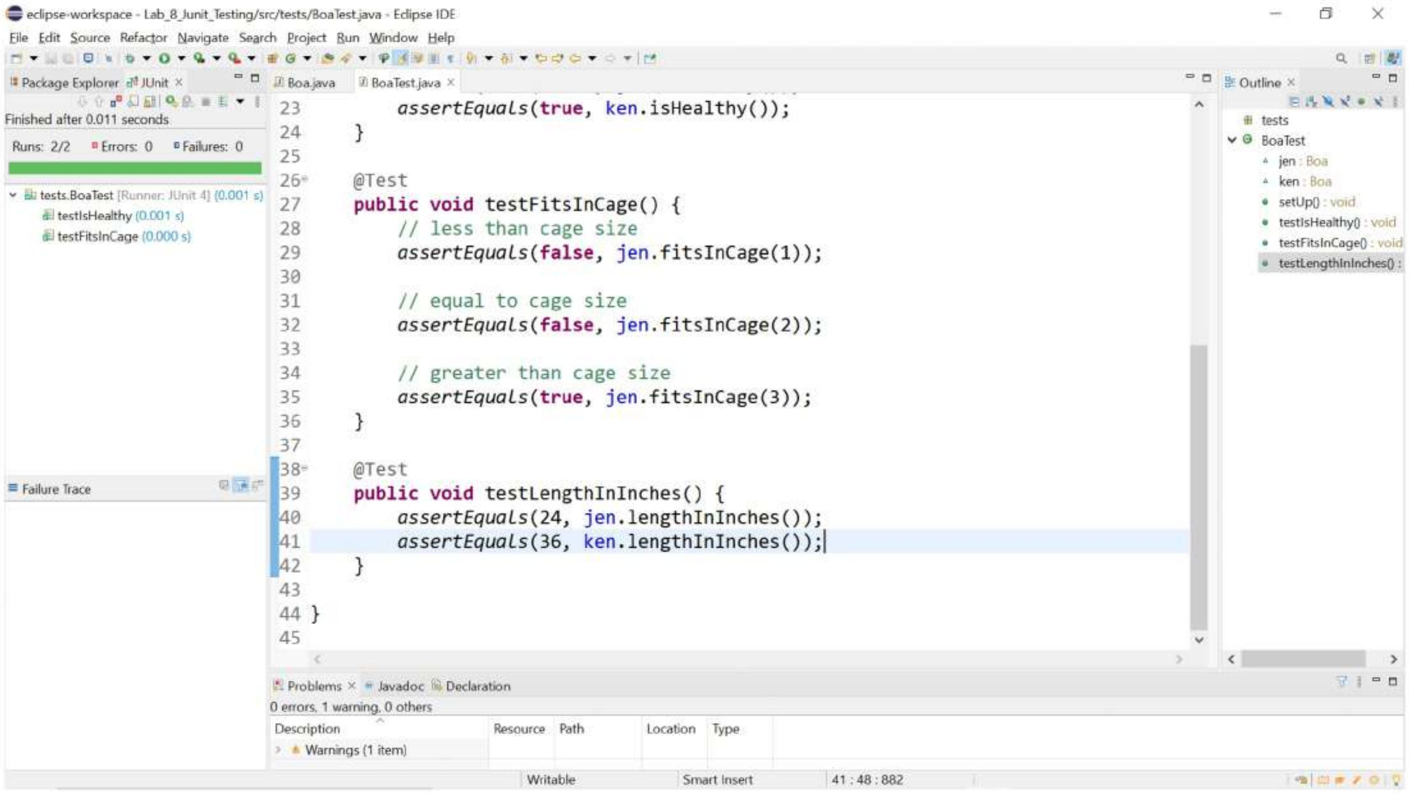
eclipse-workspace - Lab_8_Junit_Testing/src/tests/Boa.java - Eclipse IDE

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

```java
    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;
    }

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

0 errors, 1 warning, 0 others
Warnings (1 item)
Writable Smart Insert 27 : 29 : 760

The Boa's length is specified in feet, so I multiplied length by 12 to convert it to inches and then returned the result.
Now, writing a new test case for testing the length in inches,



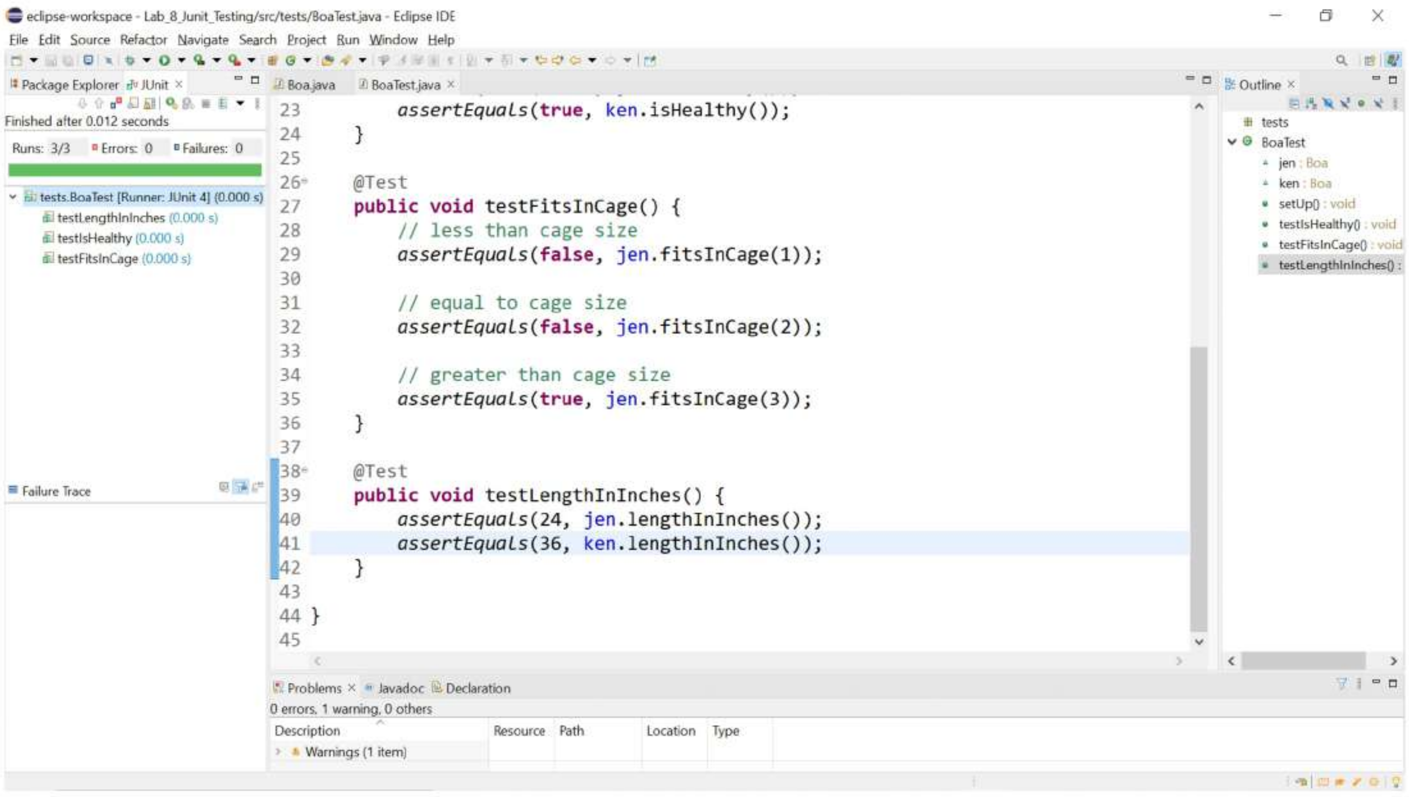
eclipse-workspace - Lab_8_Junit_Testing/src/tests/BoaTest.java - Eclipse IDE

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

```java
        assertEquals(true, ken.isHealthy());
    }

    @Test
    public void testFitsInCage() {
        // less than cage size
        assertEquals(false, jen.fitsInCage(1));

        // equal to cage size
        assertEquals(false, jen.fitsInCage(2));

        // greater than cage size
        assertEquals(true, jen.fitsInCage(3));
    }

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

}
```

0 errors, 1 warning, 0 others
Warnings (1 item)
Writable Smart Insert 41 : 48 : 882

Running the newly created test cases,


eclipse-workspace - Lab_8_Junit_Testing/src/tests/BoaTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
Boa.java
BoaTest.java
Outline
Failure Trace
Problems
Javadoc
Declaration

```
Finished after 0.012 seconds
Runs: 3/3    Errors: 0    Failures: 0

tests.BoaTest [Runner: JUnit 4] (0.000 s)
    testLengthInInches (0.000 s)
    testIsHealthy (0.000 s)
    testFitsInCage (0.000 s)
```

```java
        assertEquals(true, ken.isHealthy());
    }

    @Test
    public void testFitsInCage() {
        // less than cage size
        assertEquals(false, jen.fitsInCage(1));

        // equal to cage size
        assertEquals(false, jen.fitsInCage(2));

        // greater than cage size
        assertEquals(true, jen.fitsInCage(3));
    }

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

}
```

```
tests
BoaTest
    jen : Boa
    ken : Boa
    setUp() : void
    testIsHealthy() : void
    testFitsInCage() : void
    testLengthInInches() :
```

```
0 errors, 1 warning, 0 others
Description          Resource   Path   Location   Type
> Warnings (1 item)
```

As a result, test cases have been created for the specified Boa class, and the necessary Junit test cases have been used to test all three methods.

END OF ASSIGNMENT