**ECSE 428 – Software Engineering Practice**

**Assignment B: Test Driven Development**

**Winter 2014**

Dao, Nhat-Quang 260457711

Singzon, Ryan 260397455

Contents

[Domain Logic Tests 2](#_Toc380933509)

[1. The layout contains any number of elements 2](#_Toc380933510)

[2. Three input fields exist 3](#_Toc380933511)

[3. Submit button exists 5](#_Toc380933512)

[User Error Tests 7](#_Toc380933513)

[1. None of the inputs are filled 7](#_Toc380933514)

[2. The input type is invalid 9](#_Toc380933515)

[3. The input is out of bounds 10](#_Toc380933516)

[4. The input does not create a triangle 11](#_Toc380933517)

[Triangle Business Rules 12](#_Toc380933518)

[1. Isosceles 12](#_Toc380933519)

[2. Equilateral 13](#_Toc380933520)

[3. Scalene 14](#_Toc380933521)

# Domain Logic Tests

## The layout contains any number of elements

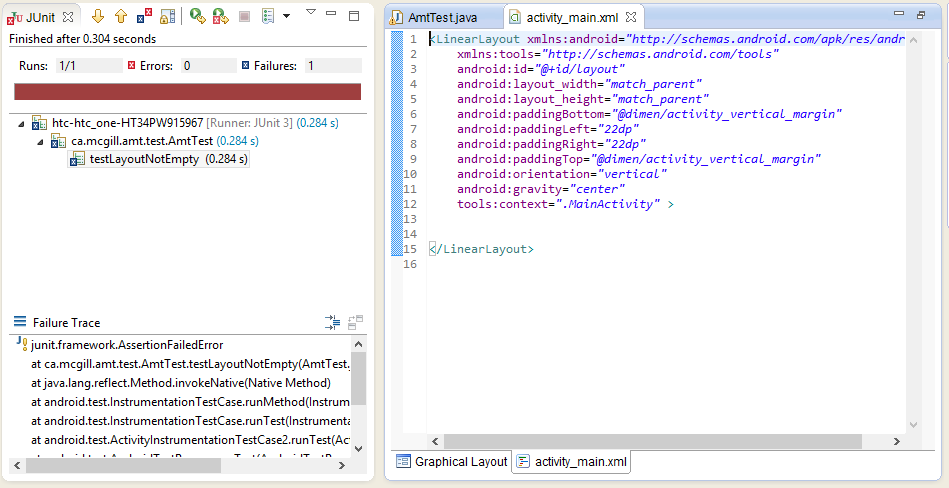
Check if the application is properly formatted by verifying that that elements appear on screen

Setup: None

Expected Result: The layout should have elements in which the user can input their

specifications

Failing Screenshot:



Code added

<EditText

android:id=*"@+id/first\_side"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

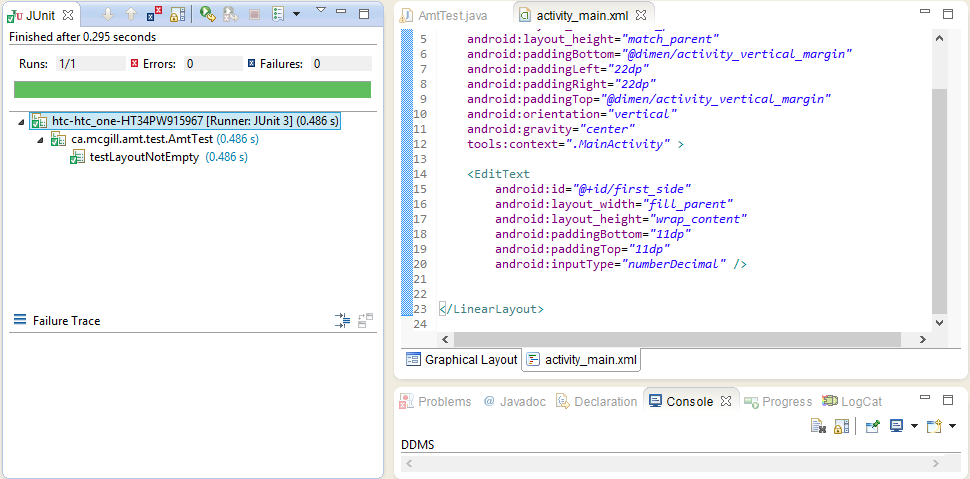
android:paddingBottom=*"11dp"*

android:paddingTop=*"11dp"*

android:hint=*"@+string/enter\_first\_side"*

android:inputType=*"numberDecimal"* />

Passing Screenshot

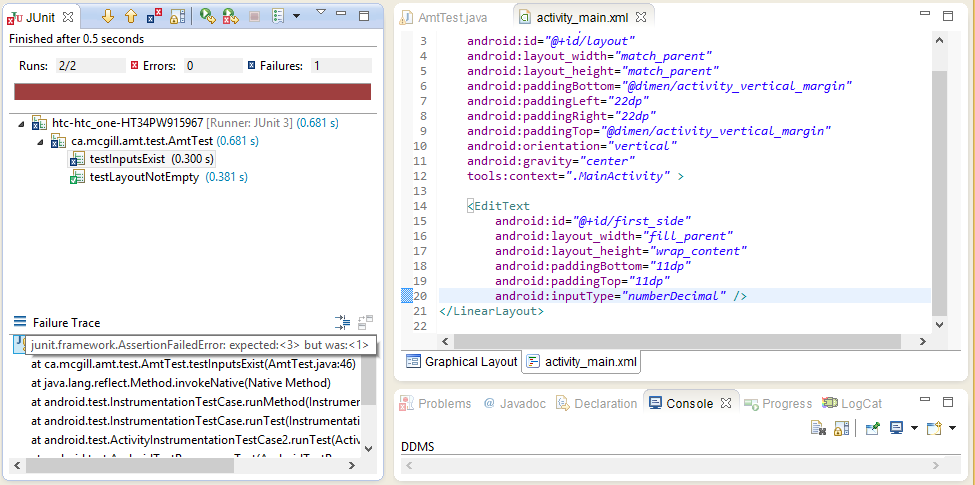


## Three input fields exist

Verify that the input fields for the user exist

Setup: None

Expected result: The layout should contain three input fields, one for each side of the triangle

Failing Screenshot:

Code added:

<EditText

android:id=*"@+id/second\_side"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:paddingBottom=*"11dp"*

android:paddingTop=*"11dp"*

android:hint=*"@+string/enter\_second\_side"*

android:inputType=*"numberDecimal"* />

<EditText

android:id=*"@+id/third\_side"*

android:layout\_width=*"fill\_parent"*

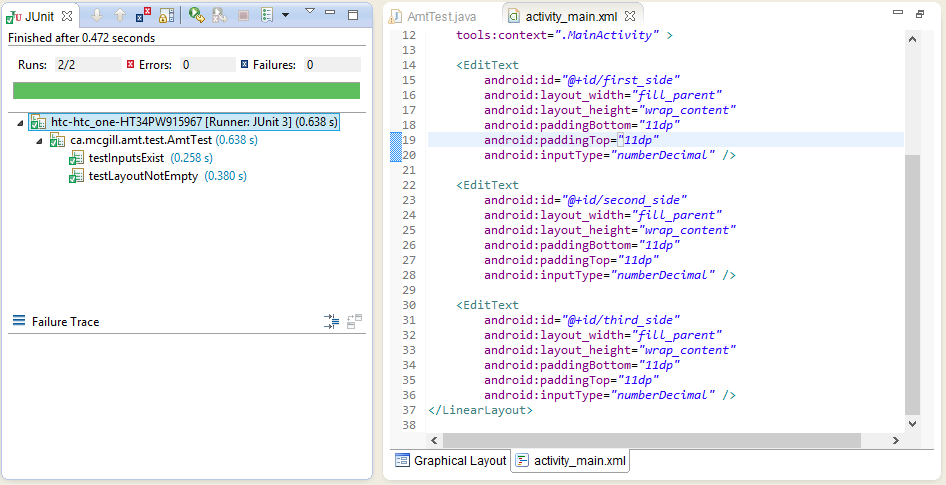
android:layout\_height=*"wrap\_content"*

android:paddingBottom=*"11dp"*

android:paddingTop=*"11dp"*

android:hint=*"@+string/enter\_third\_side"*

android:inputType=*"numberDecimal"* />

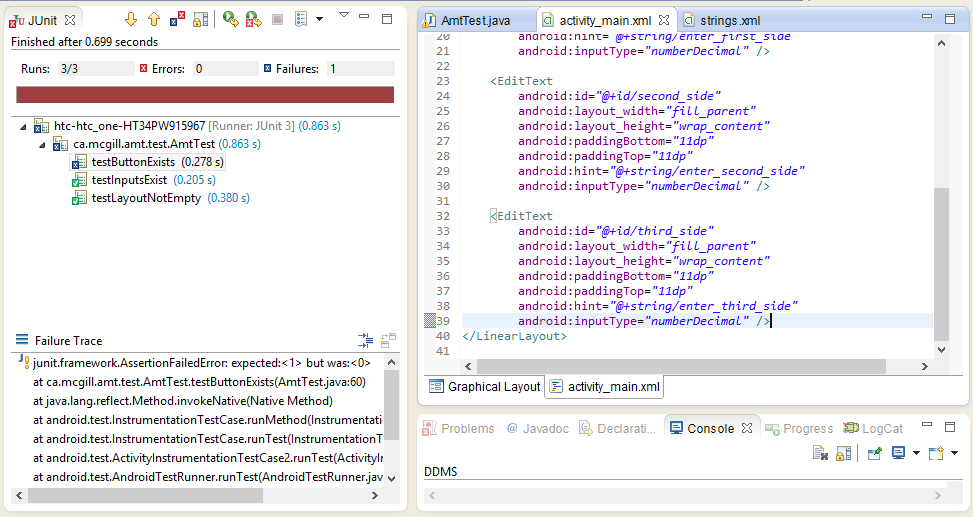
 Passing screenshot:

## Submit button exists

Setup: None

Expected result: The activity should display a button that a user presses to obtain a result for their input

Failing screenshot:



Code added:

<Button

android:id=*"@+id/submit"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

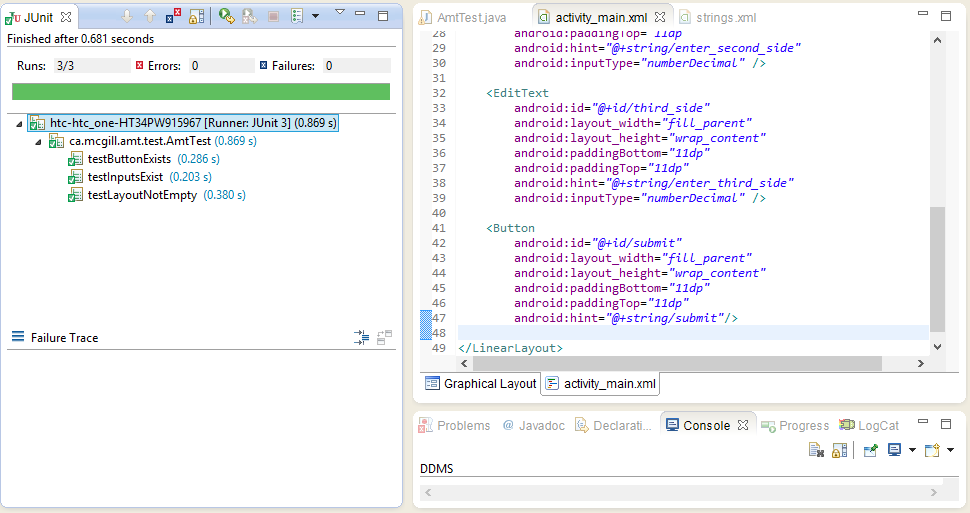
android:paddingBottom=*"11dp"*

android:paddingTop=*"11dp"*

android:hint=*"@+string/submit"*

android:onClick=*"getTriangle"*/>

Passing screenshot:



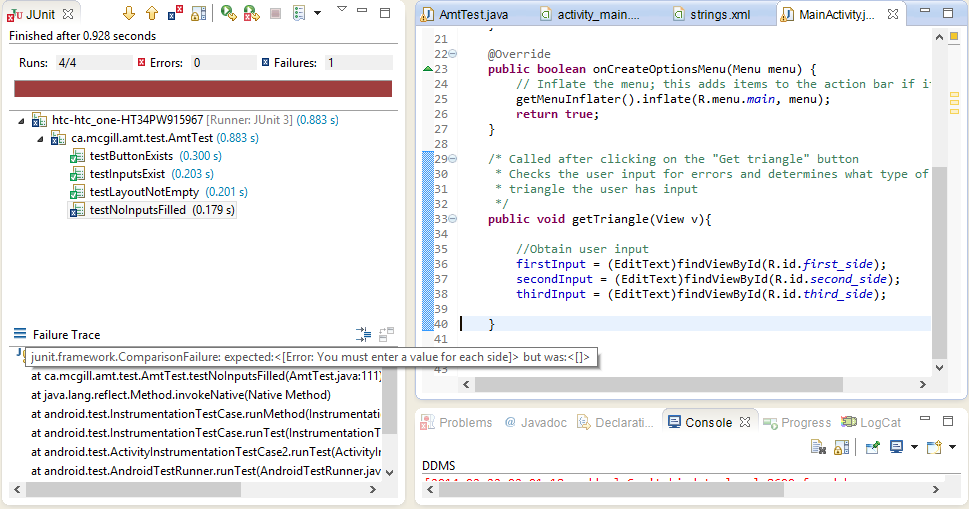
# User Error Tests

## None of the inputs are filled

Setup: Each input is left blank, then the submit button is pressed

Expected result: An error message appears telling the user that all inputs must be filled

Failing screenshot:



Code added:

//Check if any of the inputs are empty

**if**(firstInput.getText().toString().equals("") ||

secondInput.getText().toString().equals("") ||

thirdInput.getText().toString().equals("") ){

//Displays an error message if any inputs are empty

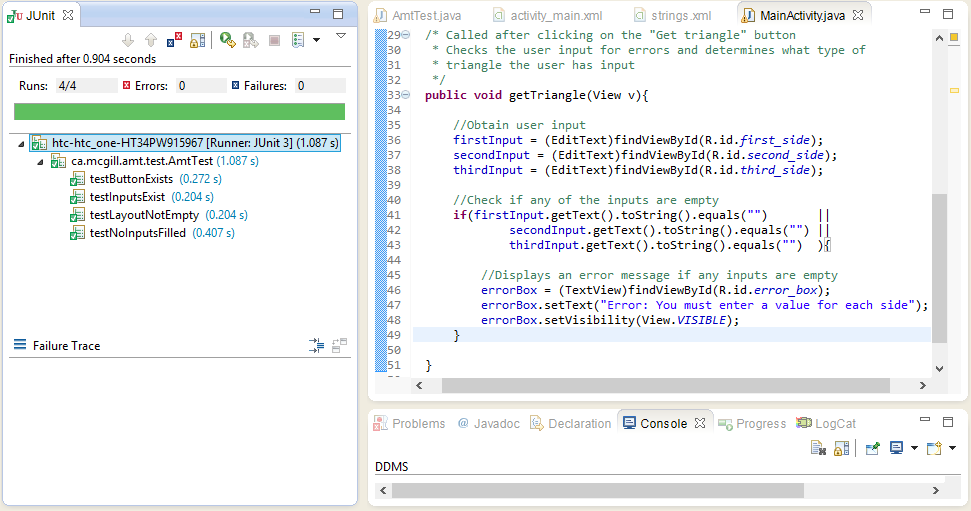
outputBox.setText("Error: You must enter a value for each side");

outputBox.setVisibility(View.*VISIBLE*);

**return**;

}

Passing Screenshot:



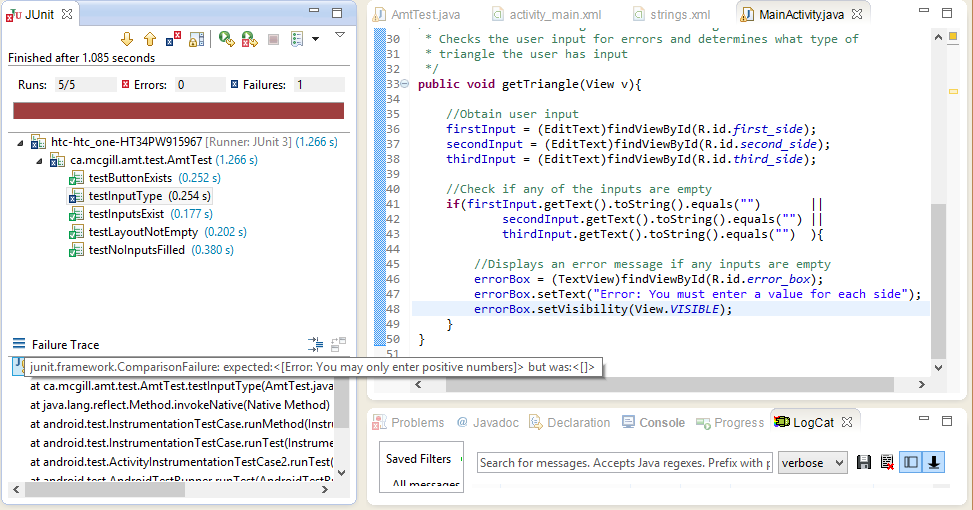
## The input type is invalid

Verify that the user can only enter whole numbers

Setup: Input characters and negative numbers into the text fields

Expected Result: An error message appears telling the user they must only enter whole numbers

Failing screenshot:



Code added:

**try** {

firstSide = Integer.*parseInt*(firstInput.getText().toString());

secondSide = Integer.*parseInt*(secondInput.getText().toString());

thirdSide = Integer.*parseInt*(thirdInput.getText().toString());

} **catch** (NumberFormatException e) {

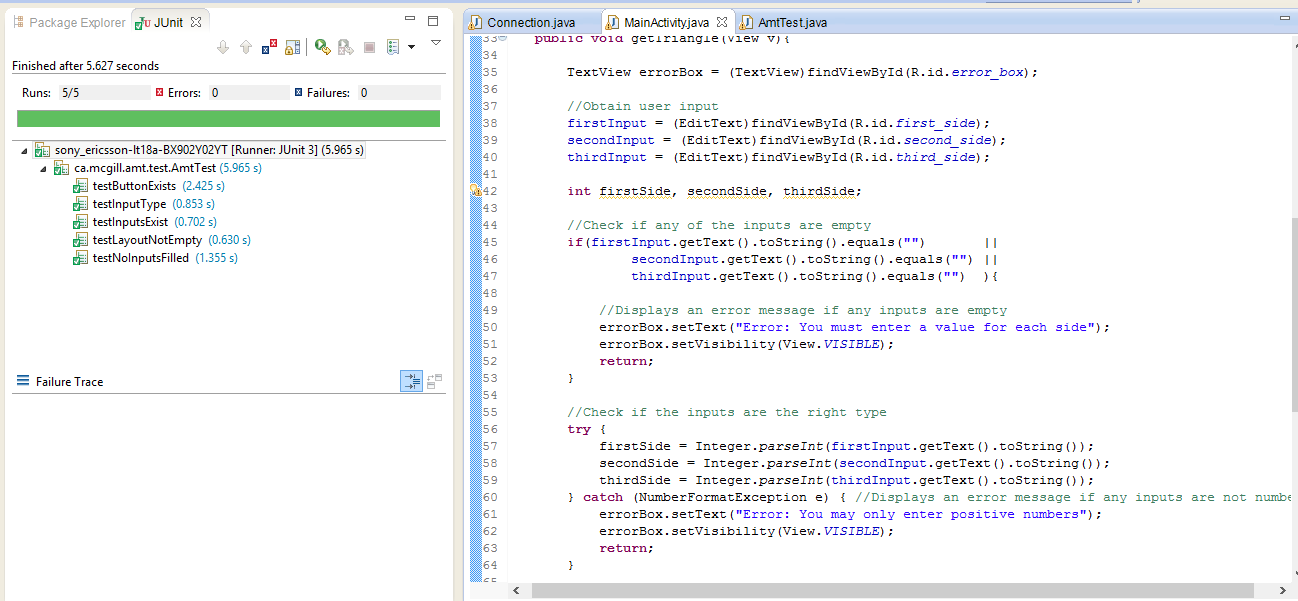
//Displays an error message if any inputs are not numbers

outputBox.setText("Error: You may only enter positive numbers");

outputBox.setVisibility(View.*VISIBLE*);

}

Passing screenshot:



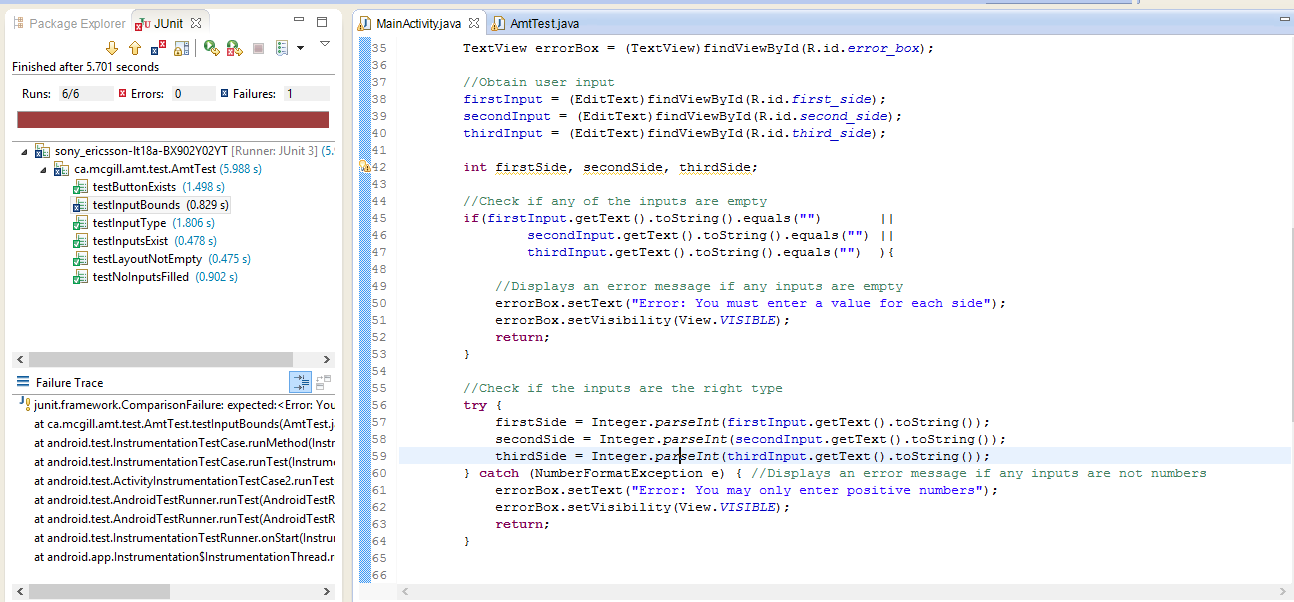
## The input is out of bounds

The user must input numbers between 1 and 100 inclusive

Setup: Set values to numbers outside of the valid range

Expected Result: A message appears telling the user that they must only enter numbers between 1 and 100 inclusive

Failing screenshot:



Code added:

//Check if the inputs are within the valid range

**if**(firstSide < 1 || firstSide > 100 ||

secondSide < 1 || secondSide > 100 ||

thirdSide < 1 || thirdSide > 100 ) {

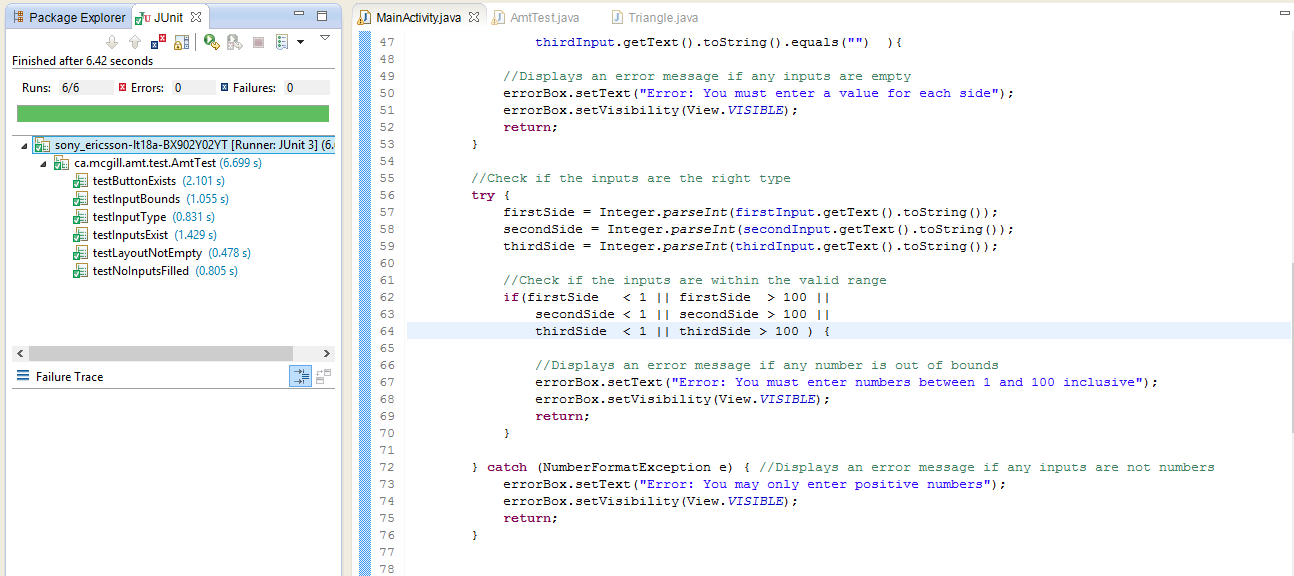
//Displays an error message if any number is out of bounds

outputBox.setText("Error: You must enter numbers between 1 and 100 inclusive");

outputBox.setVisibility(View.*VISIBLE*);

}

Passing screenshot:



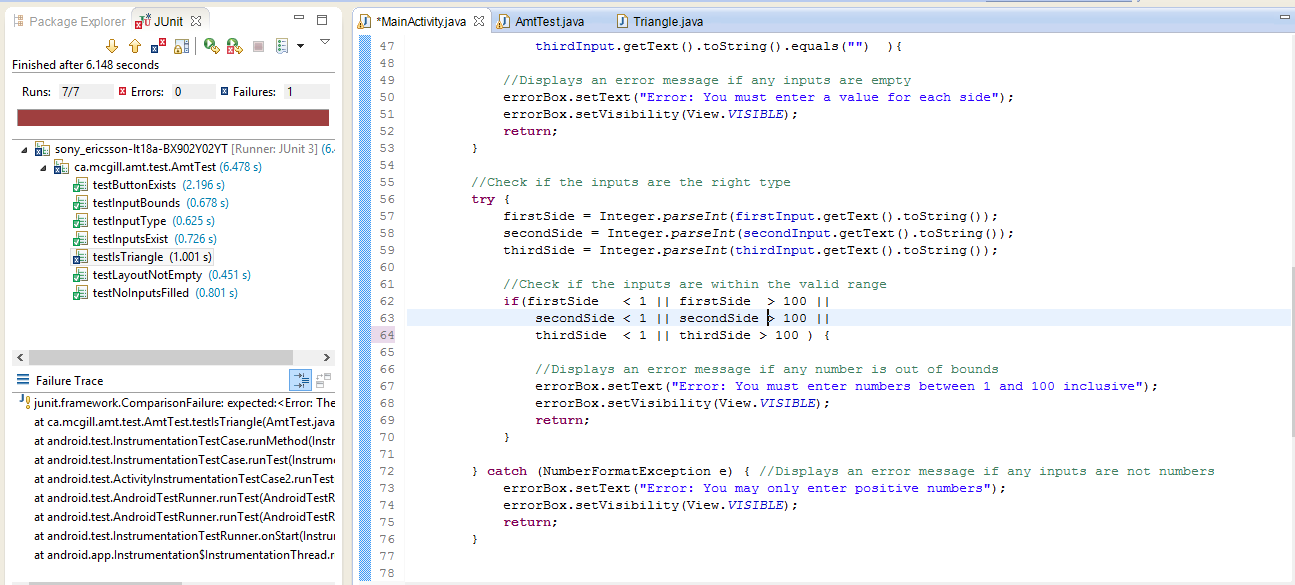
## The input does not create a triangle

The values of the sides the user entered does not create a triangle

Setup: Set two of the inputs such that their sum is smaller than the third input

Expected Result: An error message will appear telling the user that their inputs do not create a triangle

Failing Screenshot:



Code added:

//Check if the inputs actually create a triangle

**else** **if**((firstSide + secondSide) < thirdSide ||

(firstSide + thirdSide ) < secondSide ||

(secondSide + thirdSide) < firstSide ){

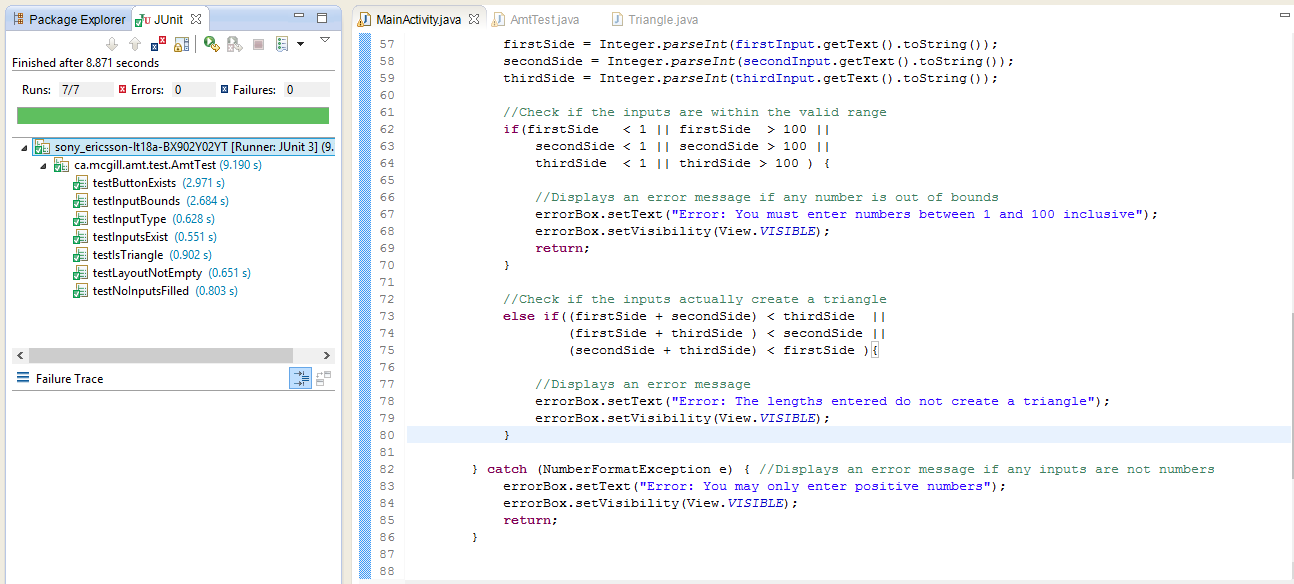
//Displays an error message

outputBox.setText("Error: The lengths entered do not create a triangle");

outputBox.setVisibility(View.*VISIBLE*);

}

Passing Screenshot:



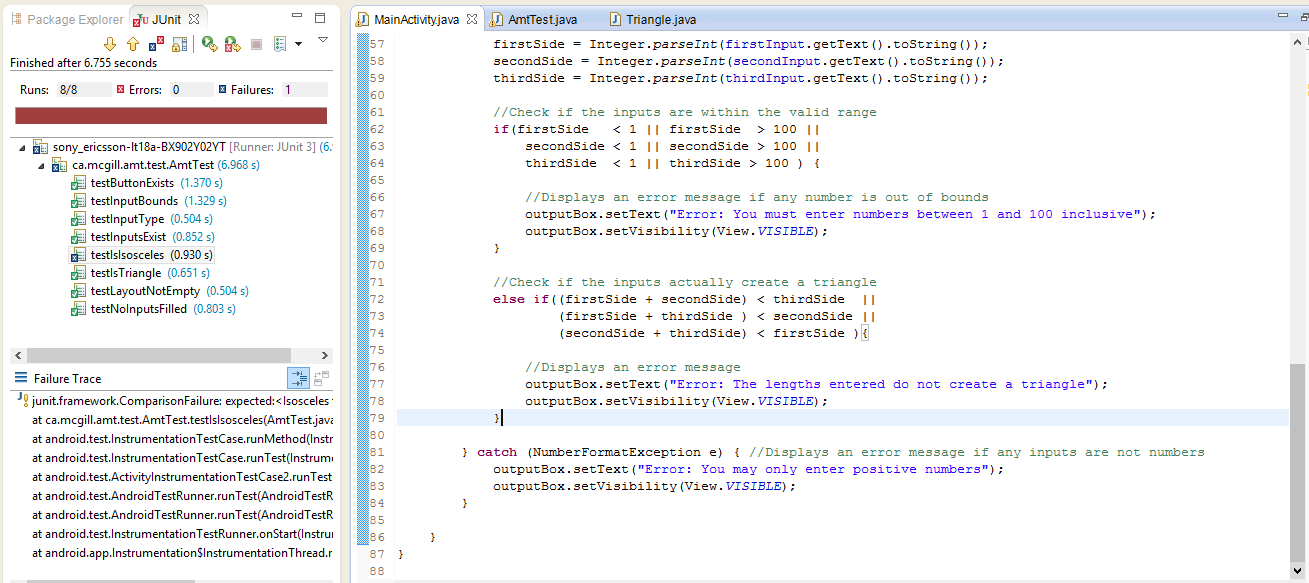
# Triangle Business Rules

## Isosceles

The user entered the same value for 2 sides.  
Set up: Set 2 inputs to have the same value.

Expected result: A message will appear telling the user that the inputs form an isosceles triangle.

Failing screenshot:



Code added:

**else** **if** (firstSide == secondSide ||

firstSide == thirdSide ||

secondSide == thirdSide) {

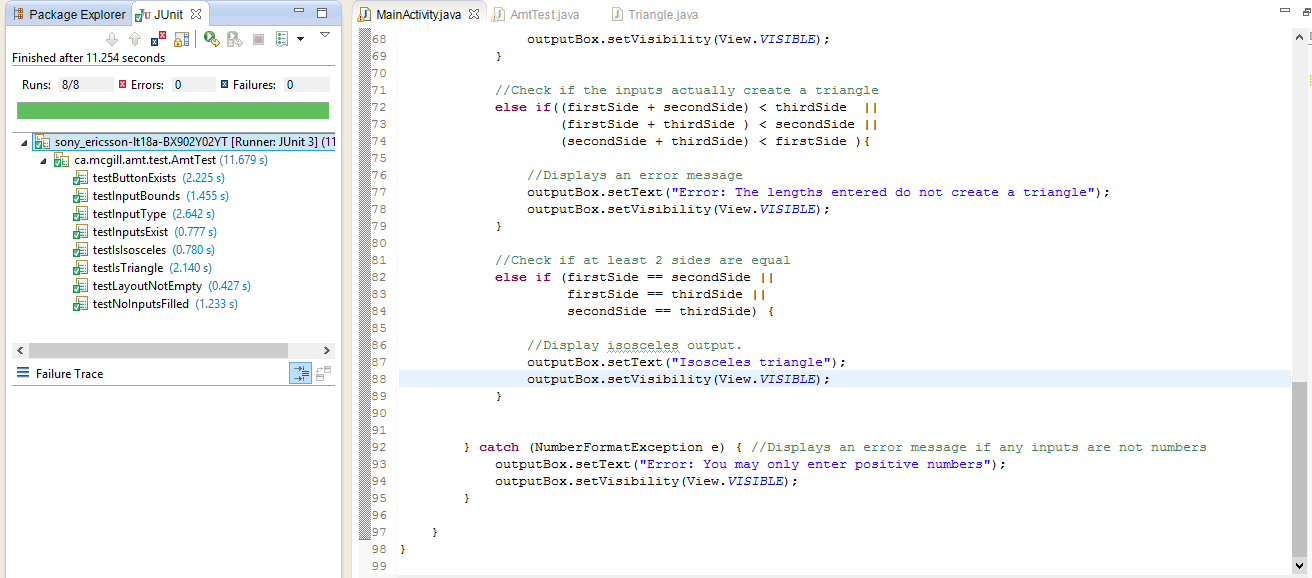
//Display output isosceles triangle

outputBox.setText("Isosceles triangle");

outputBox.setVisibility(View.*VISIBLE*);

}

Passing screenshot:

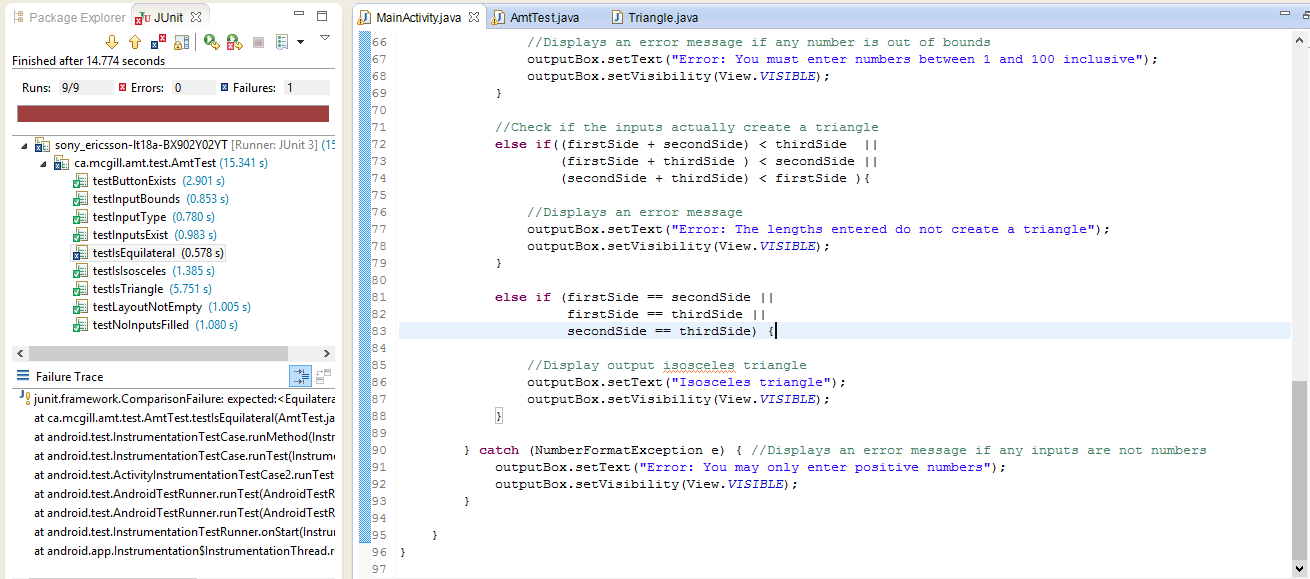


## Equilateral

The user entered the same value for all three sides.  
Set up: Set all three inputs to have the same value.

Expected result: A message will appear telling the user that the inputs form an equilateral triangle.

Failing screenshot:



Code added:

//Check if all inputs are equal.

**if** (firstSide == secondSide && firstSide == thirdSide) {

//Display output equilateral triangle

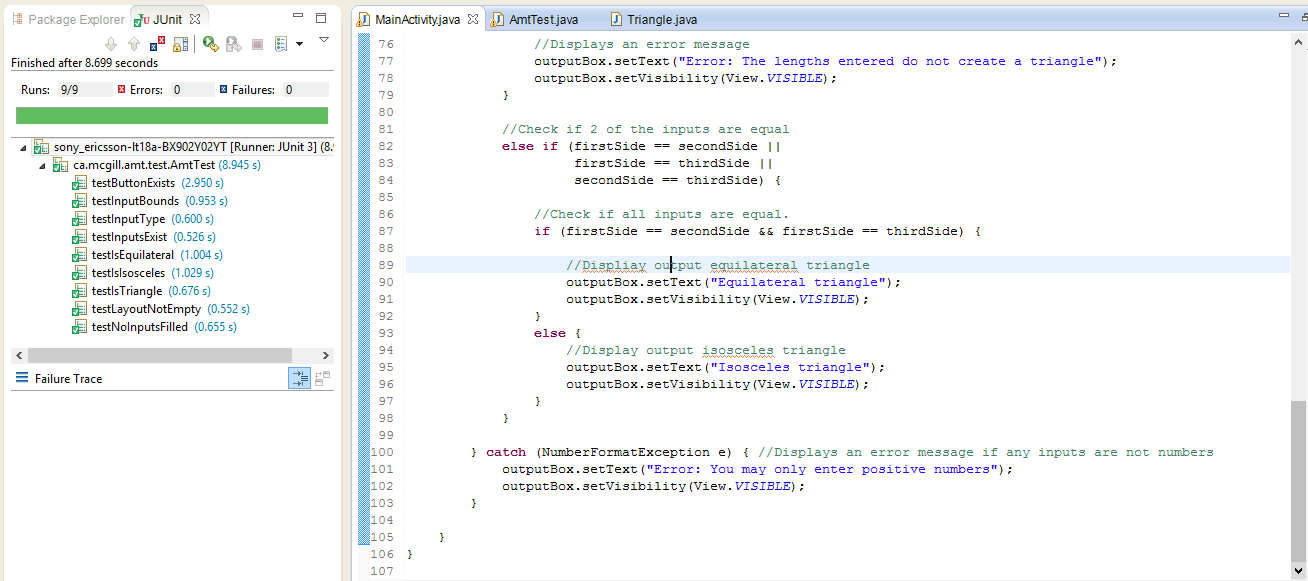
outputBox.setText("Equilateral triangle");

outputBox.setVisibility(View.*VISIBLE*);

}

**else** {

Passing screenshot:



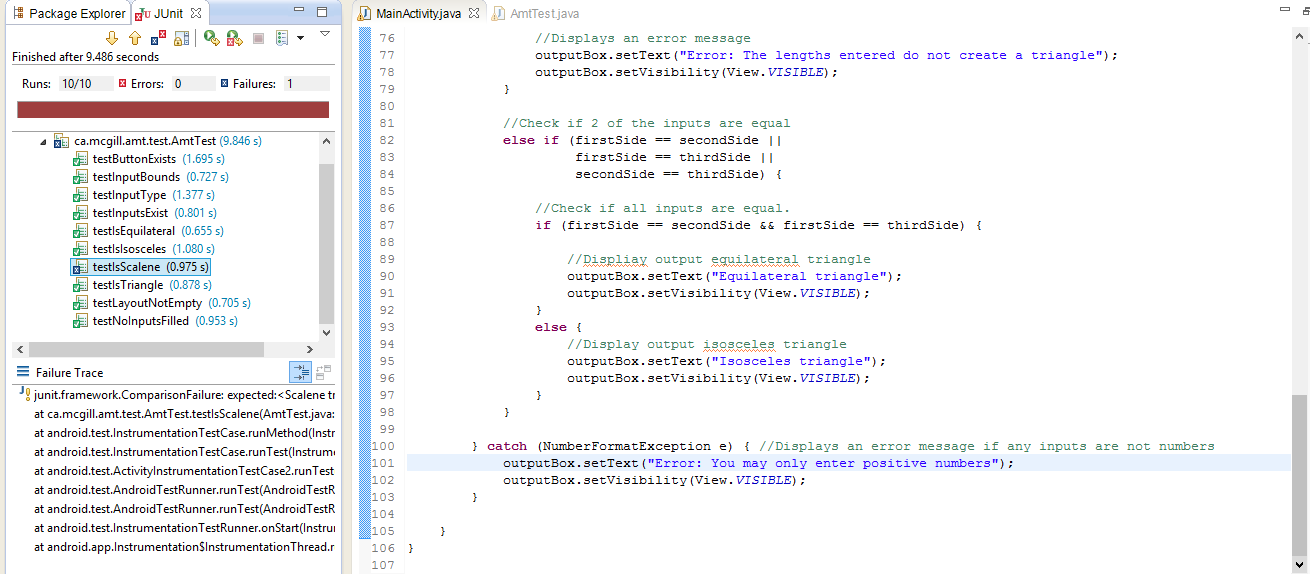
## Scalene

The user entered 3 different values for 3 sides.

Set up: Set 3 inputs to have different values.

Expected result: A message will appear telling the user that the inputs form a scalene triangle.

Failing screenshot:



Code added:

**else** {

//Display output scalene triangle

outputBox.setText("Scalene triangle");

outputBox.setVisibility(View.*VISIBLE*);

}

Passing screenshot:

