PROG8171

Programming: SQA Techniques \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Assignment #2 – Unit Testing

**75 Marks – worth 7.5%**

**Submitted By**

|  |  |  |  |
| --- | --- | --- | --- |
| First Name | Last Name | Student ID | Date |
| Pretty | Antony | 8935790 | March-06-2024 |

1. **Copy of the test source code**

*using TriangleSolver;*

*namespace TriangleSolverTest*

*{*

*public class Tests*

*{*

*//tests for Equilateral triangle*

*[Test]*

*public void AnalyzeTriangle\_Input30and30and30\_OutputValidEquilateralTriangle()*

*{*

*//Arrange*

*int firstSide = 30;*

*int secondSide = 30;*

*int thirdSide = 30;*

*string expectedResponse = "An equilateral triangle is formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*//tests for Isosceles triangle*

*[Test]*

*public void AnalyzeTriangle\_Input30and30and25\_OutputValidIsoscelesTriangle()*

*{*

*//Arrange*

*int firstSide = 30;*

*int secondSide = 30;*

*int thirdSide = 25;*

*string expectedResponse = "An isosceles triangle is formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input30and35and35\_OutputValidIsoscelesTriangle()*

*{*

*//Arrange*

*int firstSide = 30;*

*int secondSide = 35;*

*int thirdSide = 35;*

*string expectedResponse = "An isosceles triangle is formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input40and35and40\_OutputValidIsoscelesTriangle()*

*{*

*//Arrange*

*int firstSide = 40;*

*int secondSide = 35;*

*int thirdSide = 40;*

*string expectedResponse = "An isosceles triangle is formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*//tests for Scalene triangle*

*[Test]*

*public void AnalyzeTriangle\_Input30and35and40\_OutputValidScaleneTriangle()*

*{*

*//Arrange*

*int firstSide = 30;*

*int secondSide = 35;*

*int thirdSide = 40;*

*string expectedResponse = "A scalene triangle is formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide,thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input20and22and40\_OutputValidScaleneTriangle()*

*{*

*//Arrange*

*int firstSide = 20;*

*int secondSide = 22;*

*int thirdSide = 40;*

*string expectedResponse = "A scalene triangle is formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input10and12and18\_OutputValidScaleneTriangle()*

*{*

*//Arrange*

*int firstSide = 10;*

*int secondSide = 12;*

*int thirdSide = 18;*

*string expectedResponse = "A scalene triangle is formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input8and9and15\_OutputValidScaleneTriangle()*

*{*

*//Arrange*

*int firstSide = 8;*

*int secondSide = 9;*

*int thirdSide = 15;*

*string expectedResponse = "A scalene triangle is formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input99and88and111\_OutputValidScaleneTriangle()*

*{*

*//Arrange*

*int firstSide = 99;*

*int secondSide = 88;*

*int thirdSide = 111;*

*string expectedResponse = "A scalene triangle is formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*//tests for Invalid triangle - one side zero*

*[Test]*

*public void AnalyzeTriangle\_Input0and8and11\_OutputOneSideIsZero()*

*{*

*//Arrange*

*int firstSide = 0;*

*int secondSide = 8;*

*int thirdSide = 11;*

*string expectedResponse = "At least one side entered had a zero - invalid triangle";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input9and0and11\_OutputOneSideIsZero()*

*{*

*//Arrange*

*int firstSide = 9;*

*int secondSide = 0;*

*int thirdSide = 11;*

*string expectedResponse = "At least one side entered had a zero - invalid triangle";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input9and12and0\_OutputOneSideIsZero()*

*{*

*//Arrange*

*int firstSide = 9;*

*int secondSide = 12;*

*int thirdSide = 0;*

*string expectedResponse = "At least one side entered had a zero - invalid triangle";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input9and10and50\_OutputInvalidTraingle()*

*{*

*//Arrange*

*int firstSide = 9;*

*int secondSide = 10;*

*int thirdSide = 50;*

*string expectedResponse = "A triangle cannot be formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input9and75and15\_OutputInvalidTriangle()*

*{*

*//Arrange*

*int firstSide = 9;*

*int secondSide = 75;*

*int thirdSide = 15;*

*string expectedResponse = "A triangle cannot be formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

*[Test]*

*public void AnalyzeTriangle\_Input90and10and50\_OutputInvalidTriangle()*

*{*

*//Arrange*

*int firstSide = 90;*

*int secondSide = 10;*

*int thirdSide = 50;*

*string expectedResponse = "A triangle cannot be formed";*

*//Act*

*var actualResposne = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);*

*//Asserts*

*Assert.IsNotEmpty(actualResposne);*

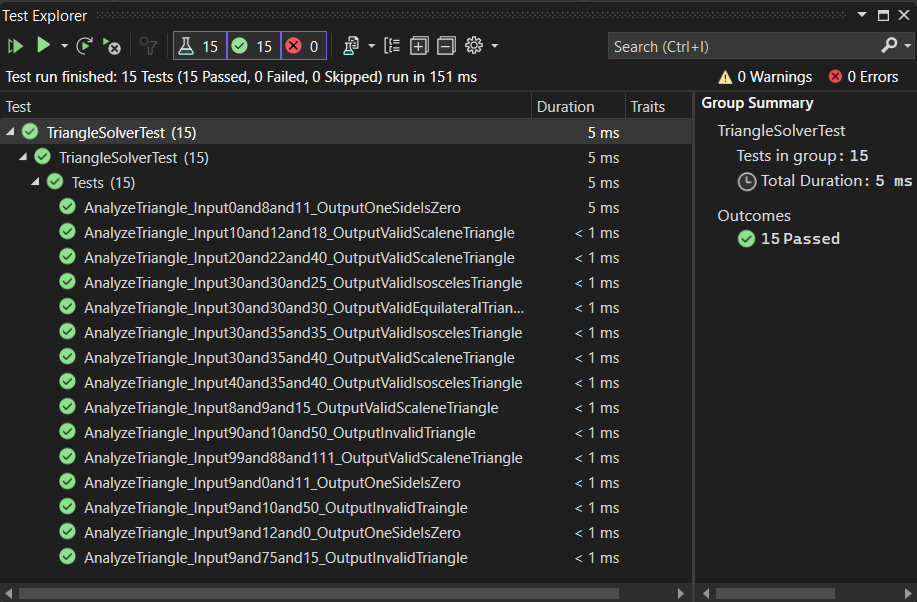
*Assert.AreEqual(actualResposne, expectedResponse);*

*}*

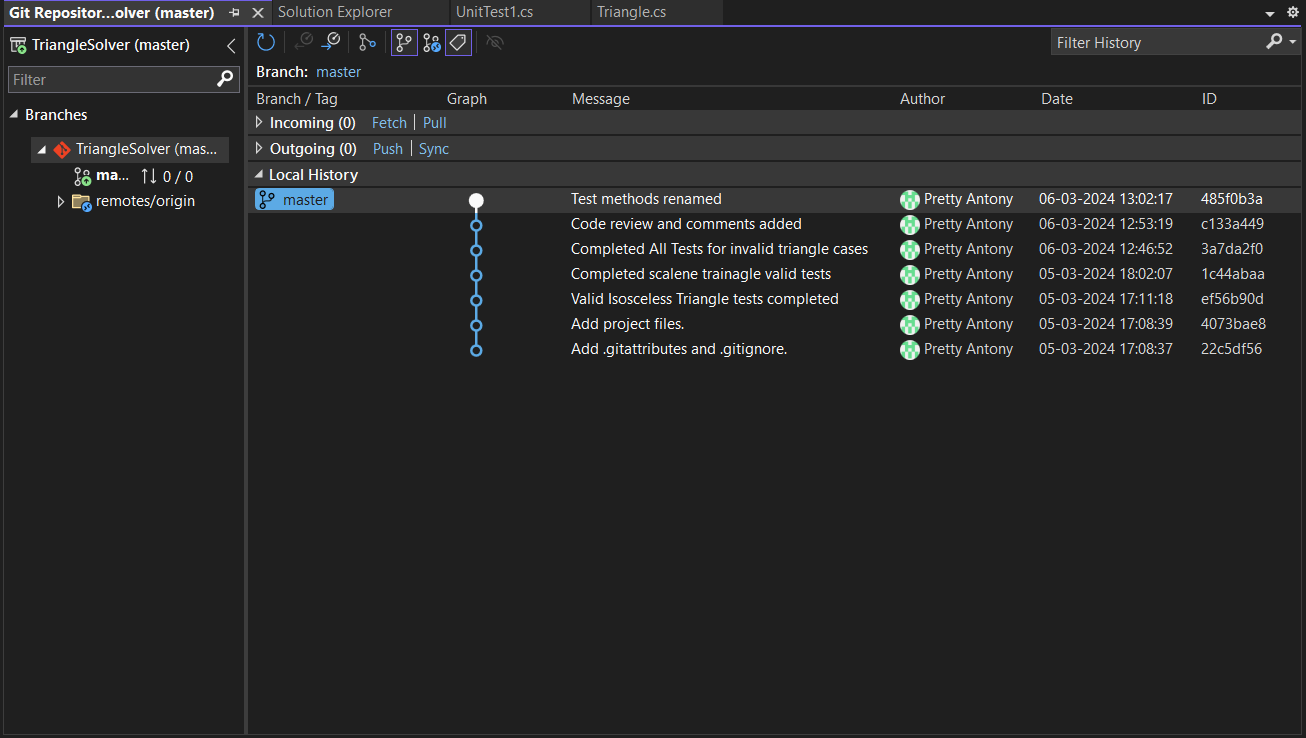
*}*

*}*

1. **Screenshot showing the results of the unit tests being run in Test Explorer**

****

1. **Screenshot/output of the GitHub repository log showing all commits**

****