Assignment 2

Name: Siyu Liu

Student ID: 8859412

Date: 2023-06-02

# Test Source Code

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using TriangleSolver;

using NUnit.Framework;

namespace TestClass

{

[TestFixture]

public class TriangleTest

{

// Equilateral Triangle Testings

[Test]

public void EquilateralTriangleTest\_Input5and5and5\_OutputEquilateral()

{

int firstSide = 5;

int secondSide = 5;

int thirdSide = 5;

string expected = "An EQUILATERAL triangle is formed";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

// Isosceles Triangle Testings

[Test]

public void IsoscelesTriangleTest\_Input5and4and4\_OutputIsosceles()

{

int firstSide = 5;

int secondSide = 4;

int thirdSide = 4;

string expected = "An ISOSCELES triangle is formed";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

[Test]

public void IsoscelesTriangleTest\_Input5and3and3\_OutputIsosceles()

{

int firstSide = 5;

int secondSide = 3;

int thirdSide = 3;

string expected = "An ISOSCELES triangle is formed";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

[Test]

public void IsoscelesTriangleTest\_Input4and5and5\_OutputIsosceles()

{

int firstSide = 4;

int secondSide = 5;

int thirdSide = 5;

string expected = "An ISOSCELES triangle is formed";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

// Scalene Triangle Testings

[Test]

public void ScaleneTriangleTest\_Input5and4and3\_OutputScalene()

{

int firstSide = 5;

int secondSide = 4;

int thirdSide = 3;

string expected = "A SCALENE triangle is formed";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

[Test]

public void ScaleneTriangleTest\_Input6and4and3\_OutputScalene()

{

int firstSide = 6;

int secondSide = 4;

int thirdSide = 3;

string expected = "A SCALENE triangle is formed";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

[Test]

public void ScaleneTriangleTest\_Input6and5and3\_OutputScalene()

{

int firstSide = 6;

int secondSide = 5;

int thirdSide = 3;

string expected = "A SCALENE triangle is formed";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

[Test]

public void ScaleneTriangleTest\_Input6and5and4\_OutputScalene()

{

int firstSide = 6;

int secondSide = 5;

int thirdSide = 4;

string expected = "A SCALENE triangle is formed";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

[Test]

public void ScaleneTriangleTest\_Input7and5and4\_OutputScalene()

{

int firstSide = 7;

int secondSide = 5;

int thirdSide = 4;

string expected = "A SCALENE triangle is formed";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

// Zero Sides Triangle Testings

[Test]

public void ZeroSidesTriangleTest\_Input5and4and0\_OutputInvalid()

{

int firstSide = 5;

int secondSide = 4;

int thirdSide = 0;

string expected = "Invalid Triangle - at least one side is zero";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

[Test]

public void ZeroSidesTriangleTest\_Input5and0and0\_OutputInvalid()

{

int firstSide = 5;

int secondSide = 0;

int thirdSide = 0;

string expected = "Invalid Triangle - at least one side is zero";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

[Test]

public void ZeroSidesTriangleTest\_Input0and0and0\_OutputInvalid()

{

int firstSide = 0;

int secondSide = 0;

int thirdSide = 0;

string expected = "Invalid Triangle - at least one side is zero";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

// Invaild Sides Triangle Testings

[Test]

public void InvalidSidesTriangleTest\_Input6and3and3\_OutputInvalid()

{

int firstSide = 6;

int secondSide = 3;

int thirdSide = 3;

string expected = "INVALID Triangle detected!!";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

[Test]

public void InvalidSidesTriangleTest\_Input8and4and3\_OutputInvalid()

{

int firstSide = 8;

int secondSide = 4;

int thirdSide = 3;

string expected = "INVALID Triangle detected!!";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

[Test]

public void InvalidSidesTriangleTest\_Input5and5and10\_OutputInvalid()

{

int firstSide = 5;

int secondSide = 5;

int thirdSide = 10;

string expected = "INVALID Triangle detected!!";

string actual = Triangle.AnalyzeTriangle(firstSide, secondSide, thirdSide);

Assert.AreEqual(expected, actual);

}

}

}

# Testing Result

A screenshot of a computer

Description automatically generated

# Commit History

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated