**Source Code:**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

/// <summary>

/// PROG2070 – Quality Assurance: Winter 2017

/// Assignment 02

/// Yoonsuk Cho Feb 14, 2017

/// </summary>

namespace YChoAssignment02

{

/// <summary>

/// main class of assignment 2

/// </summary>

class Program

{

/// <summary>

/// program start main method

/// using TriangleSolver.Analyze

/// it will show the type of triangle with input sides

/// </summary>

/// <param name="args">parameters of start method</param>

static void Main(string[] args)

{

bool isCont = true;

int input = 0;

int[] sides = { -1, -1, -1 };

while (isCont)

{

Console.WriteLine("1.Enter triangle dimensions");

Console.WriteLine("2.Exit");

Console.WriteLine();

Console.Write("Please select number (1-2): ");

if (Int32.TryParse(Console.ReadLine(), out input))

{

if (input < 1 || input > 2) input = 3;

}

else input = 3;

switch (input)

{

// when selected '1.Enter triangle dimensions' in the menu

case 1:

Console.WriteLine("");

bool isContIn = true;

int sideOrder = 0;

while (isContIn)

{

Console.Write("Enter the side{0}: ", (sideOrder + 1));

input = sides[sideOrder];

if (Int32.TryParse(Console.ReadLine(), out input))

{

// case of less than 1

if (input < 1)

{

Console.WriteLine("Input must be greater than 0.");

Console.WriteLine();

input = -1;

}

}

else

{

// case of non integer

Console.WriteLine("Input must be an integer.");

Console.WriteLine();

}

// integer and greater than 0

if (input > 0)

{

sides[sideOrder] = input;

sideOrder++;

Console.WriteLine();

}

// all 3 inputs are integer and greater than 0

if (sideOrder > 2)

{

isContIn = false;

string message = TriangleSolver.Analyze(sides);

Console.WriteLine(message);

Console.WriteLine();

clearScreen();

}

}

break;

case 2:

// when selected '2.Exit' in the menu

isCont = false;

Console.WriteLine("Bye !");

clearScreen();

break;

default:

Console.WriteLine("You have got a wrong input!");

clearScreen();

break;

}

}

}

/// <summary>

/// show message and clear message

/// if press any key, program goes on.

/// </summary>

private static void clearScreen()

{

Console.WriteLine("Press any key !");

Console.ReadKey();

Console.Clear();

}

}

}

**TriangleSolver.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

/// <summary>

/// PROG2070 – Quality Assurance: Winter 2017

/// Assignment 02

/// Yoonsuk Cho Feb 14, 2017

/// </summary>

namespace YChoAssignment02

{

/// <summary>

/// subclass of main class

/// this class would classify the type of triangle or not

/// </summary>

public class TriangleSolver

{

private static string[] msgs = { "This is a scalene triangle.",

"This is an isosceles triangle.",

"This is an equilateral triangle.",

"This is not a triangle." };

/// <summary>

/// Analyze the triangle sides

/// using 3 sides of triangle which are parameters of method

/// this method will calculate the type of triangle

/// There are 4 type of messages

/// 1. This is not a triangle.

/// 2. This is a scalene triangle.

/// 3. This is an isosceles triangle.

/// 4. This is an equilateral triangle.

/// </summary>

/// <param name="sides">3 sides of triangle</param>

/// <returns>result type of triangle</returns>

public static string Analyze(int[] sides)

{

string message = "";

// one side is bigger or eaual to sum of other 2 sides

if (sides[0] >= sides[1] + sides[2])

{

// not a triangle

message = msgs[3];

Console.WriteLine("No!");

}

// one side is bigger or eaual to sum of other 2 sides

else if (sides[1] >= sides[2] + sides[0])

{

// not a triangle

message = msgs[3];

Console.WriteLine("No!");

}

// one side is bigger or eaual to sum of other 2 sides

else if (sides[2] >= sides[0] + sides[1])

{

// not a triangle

message = msgs[3];

Console.WriteLine("No!");

}

else

{

Console.WriteLine("Yes!");

// all 3 sides are same

if (sides[0] == sides[1] && sides[1] == sides[2])

{

// an equilateral triangle

message = msgs[2];

}

// 2 sides are same, but the other is different

else if (sides[0] == sides[1])

{

// an isosceles triangle

message = msgs[1];

}

else if (sides[1] == sides[2])

{

// an isosceles triangle

message = msgs[1];

}

else if (sides[0] == sides[2])

{

// an isosceles triangle

message = msgs[1];

}

else

{

// all sides are different each other

// a scalene triangle

message = msgs[0]; ;

}

}

return message;

}

}

}

**TriangleSolverTest.cs**

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using NUnit.Framework;

/// <summary>

/// PROG2070-17W-Sec4-Programming: Software Quality Assurance

/// Assignment02 - Test Program

///

/// Yoonsuk Cho #7135551

/// Feb 14, 2017

/// </summary>

namespace YChoAssignment02.Tests

{

/// <summary>

/// NUnit Test class

/// This class tests TriangleSolver class

/// </summary>

[TestClass]

public class TriangleSolverTest

{

/// <summary>

/// This method tests various case of

/// Analyze method of TriangleSolver class

/// </summary>

/// <param name="side1">one side of triangle</param>

/// <param name="side2">one side of triangle</param>

/// <param name="side3">one side of triangle</param>

/// <param name="msg">expected return message</param>

// not a triangle: The 3rd side is not greater than sum of rest 2 sides.

[TestCase(2, 2, 4, "This is not a triangle.")]

// not a triangle: The 1st side is not greater than sum of rest 2 sides.

[TestCase(5, 2, 2, "This is not a triangle.")]

// not a triangle: The 2nd side is not greater than sum of rest 2 sides.

[TestCase(2, 6, 4, "This is not a triangle.")]

// scalene triangle: All 3 sides are different each other.

[TestCase(4, 2, 3, "This is a scalene triangle.")]

// isosceles triangle: The 1st and 2nd sides are same.

[TestCase(3, 3, 4, "This is an isosceles triangle.")]

// isosceles triangle: The 3rd and 2nd sides are same.

[TestCase(3, 4, 4, "This is an isosceles triangle.")]

// isosceles triangle: The 1st and 3rd sides are same.

[TestCase(4, 3, 4, "This is an isosceles triangle.")]

// equilateral triangle: All 3 sides are same.

[TestCase(3, 3, 3, "This is an equilateral triangle.")]

public void AnalyzeTest(int side1, int side2, int side3, string msg)

{

string message = TriangleSolver.Analyze(new int[] { side1, side2, side3 });

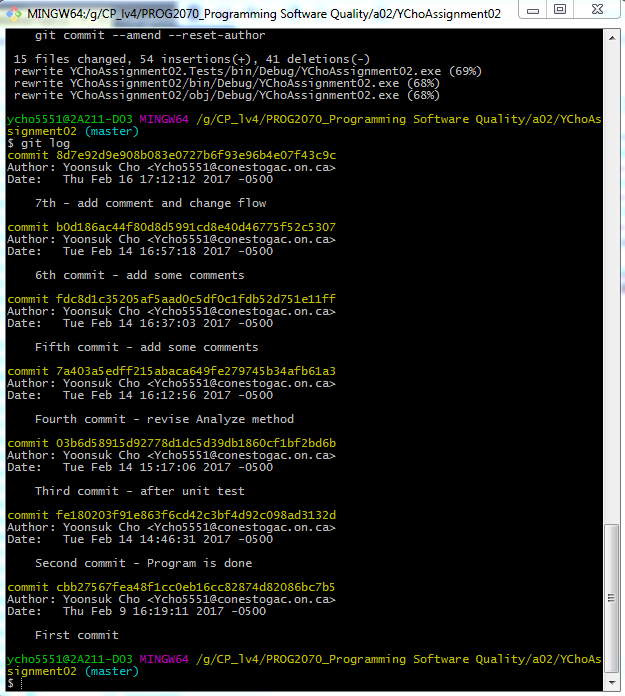
NUnit.Framework.Assert.AreEqual(msg, message);

}

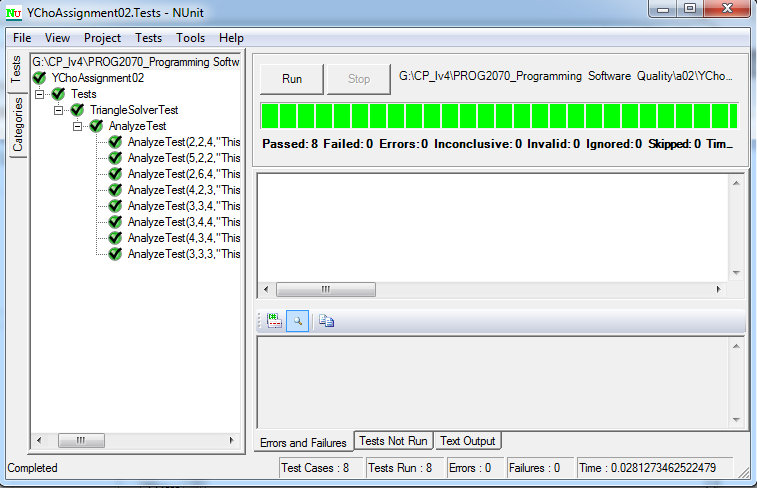
}

}

**Screen Shot of Git Bash**



**Screen Shot of Test**



**Control Flow Graphs**

