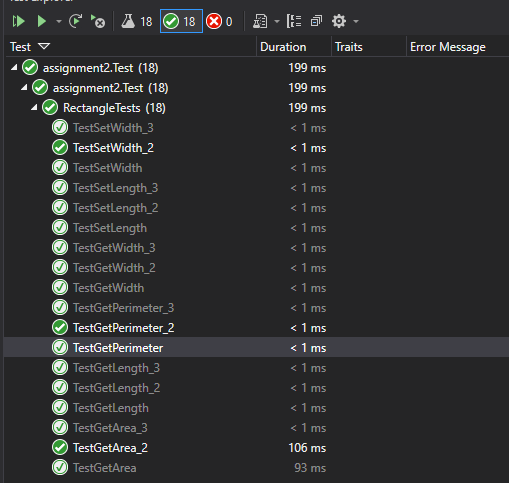
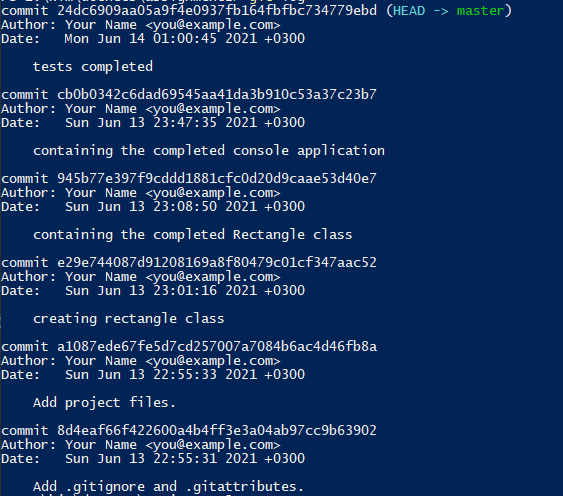
Assignment #2

Date: 14th of June 2021

Name:

Student Id:

Test Screenshot



Git Log

#### Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace assignment2

{

class Program

{

static void Main(string[] args)

{

Boolean close\_width\_prompt = true;

Boolean close\_length\_prompt = true;

int width = 0;

int length = 0;

while (close\_width\_prompt)

{

Console.Write("Enter Width : ");

string wth = Console.ReadLine();

try

{

width = Convert.ToInt32(wth);

if (width > 0)

{

close\_width\_prompt = false;

}

else

{

Console.WriteLine("\n Please enter a number greator than 0");

}

}

catch (Exception)

{

Console.WriteLine("\n Please enter a number");

}

}

while (close\_length\_prompt)

{

Console.Write("Enter Length : ");

string len = Console.ReadLine();

try

{

length = Convert.ToInt32(len);

if (length > 0)

{

close\_length\_prompt = false;

}

else

{

Console.WriteLine("\n Please enter a number greator than 0");

}

}

catch (Exception)

{

Console.WriteLine("\n Please enter a number");

}

}

Rectangle rectangle = new Rectangle(length, width);

while (true)

{

Console.WriteLine("Enter a number to perfom an action");

Console.WriteLine("1.Get Rectangle Length");

Console.WriteLine("2.Change Rectangle Length");

Console.WriteLine("3.Get Rectangle Width");

Console.WriteLine("4.Change Rectangle Width");

Console.WriteLine("5.Get Rectangle Perimeter");

Console.WriteLine("6.Get Rectangle Area");

Console.WriteLine("7.Exit");

string choice = Console.ReadLine();

try

{

int userInput = Convert.ToInt32(choice);

switch(userInput){

case 1:

Console.WriteLine(rectangle.GetLength());

break;

case 2:

int new\_length = getValue("Length");

rectangle.SetLength(new\_length);

break;

case 3:

Console.WriteLine(rectangle.GetWidth());

break;

case 4:

int new\_width = getValue("Width");

rectangle.SetLength(new\_width);

break;

case 5:

Console.WriteLine(rectangle.GetPerimeter());

break;

case 6:

Console.WriteLine(rectangle.GetArea());

break;

case 7:

Environment.Exit(0);

break;

default:

Console.WriteLine("Enter a number between 1 and 7");

break;

}

}

catch (Exception)

{

Console.WriteLine("\n Please enter a number");

}

}

}

private static int getValue(string v)

{

bool is\_active = true;

int value = 0;

while (is\_active)

{

Console.Write("Enter "+ v +" : ");

string wth = Console.ReadLine();

try

{

value = Convert.ToInt32(wth);

if (value > 0)

{

is\_active = false;

return value;

}

else

{

Console.WriteLine("\n Please enter a number greator than 0");

}

}

catch (Exception)

{

Console.WriteLine("\n Please enter a number");

}

}

return value;

}

}

}

#### Rectangle.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace assignment2

{

public class Rectangle

{

private int length, width;

public Rectangle(int length, int width)

{

// none default

this.length = length;

this.width = width;

}

public Rectangle()

{

// default

this.length = 1;

this.width = 1;

}

public int GetLength()

{

return this.length;

}

public int SetLength(int length)

{

return this.length = length;

}

public int GetWidth()

{

return this.width;

}

public int SetWidth(int width)

{

return this.width = width;

}

public int GetPerimeter()

{

int perimeter = this.width + this.length;

return perimeter \* 2;

}

public int GetArea()

{

return this.length \* this.width;

}

}

}

#### Rectangle.Test.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using assignment2;

using NUnit.Framework;

namespace assignment2.Test

{

[TestFixture]

public class RectangleTests

{

//arrange

int length = 48;

int width = 88;

//act

Rectangle rectangle = new Rectangle();

[Test]

public void TestGetLength()

{

//assert

Assert.IsNotNull(this.rectangle.GetLength());

}

[Test]

public void TestSetLength()

{

//assert

Assert.AreEqual(2,this.rectangle.SetLength(2));

}

[Test]

public void TestGetWidth()

{

//assert

Assert.IsNotNull(this.rectangle.GetWidth());

}

[Test]

public void TestSetWidth()

{

//assert

Assert.AreEqual(21, this.rectangle.SetWidth(21));

}

[Test]

public void TestGetPerimeter()

{

//assert

Assert.IsNotNull(this.rectangle.GetPerimeter());

}

[Test]

public void TestGetArea()

{

//assert

Assert.IsNotNull(this.rectangle.GetArea());

}

//End of testing default

[Test]

public void TestGetLength\_2()

{

//arrange

Rectangle newRectangle = new Rectangle(this.length, this.width);

int lens;

//act

lens = newRectangle.GetLength();

//assert

Assert.AreNotEqual(lens,this.rectangle.GetLength());

}

[Test]

public void TestSetLength\_2()

{

//arrange

int lens;

//act

lens = this.rectangle.SetLength(this.length);

//assert

Assert.AreEqual(lens, this.rectangle.GetLength());

}

[Test]

public void TestGetWidth\_2()

{

//arrange

Rectangle newRectangle = new Rectangle(this.length, this.width);

int widths;

//act

widths = newRectangle.GetLength();

//assert

Assert.AreNotEqual(widths,this.rectangle.GetWidth());

}

[Test]

public void TestSetWidth\_2()

{

//assert

Assert.AreNotEqual(this.rectangle.SetLength(this.length), this.rectangle.GetWidth());

}

[Test]

public void TestGetPerimeter\_2()

{

//arrange

int perimeter;

//act

perimeter = 2 \* (this.width + this.length);

//assert

Assert.AreNotEqual(perimeter,this.rectangle.GetPerimeter());

}

[Test]

public void TestGetArea\_2()

{

//arrange

int area;

//act

area = this.width \* this.length;

//assert

Assert.AreNotEqual(area,this.rectangle.GetArea());

}

// Final tests

[Test]

public void TestGetLength\_3()

{

//arrange

int lengths = 93;

int widths = 60;

int theLength;

Rectangle newRectangle = new Rectangle(lengths, widths);

//act

theLength = newRectangle.GetLength();

//assert

Assert.AreEqual(lengths, theLength);

}

[Test]

public void TestSetLength\_3()

{

//arrange

int lengths = 93;

int widths = 60;

int theLength;

int newLength = 8798;

Rectangle newRectangle = new Rectangle(lengths, widths);

//act

theLength = newRectangle.SetLength(newLength);

//assert

Assert.AreEqual(theLength, newLength);

}

[Test]

public void TestGetWidth\_3()

{

//arrange

int lengths = 19;

int widths = 76;

int theWidth;

Rectangle newRectangle = new Rectangle(lengths, widths);

//act

theWidth = newRectangle.GetLength();

//assert

Assert.AreNotEqual(widths, theWidth);

}

[Test]

public void TestSetWidth\_3()

{

//arrange

int lengths = 193;

int widths = 6;

int theWidths;

int newWidths = 898;

Rectangle newRectangle = new Rectangle(lengths, widths);

//act

theWidths = newRectangle.SetLength(newWidths);

//assert

Assert.AreEqual(theWidths, newWidths);

}

[Test]

public void TestGetPerimeter\_3()

{

//arrange

int perimeter;

int new\_width = 76;

int new\_height = 13;

Rectangle newRectangle = new Rectangle(new\_width, new\_height);

//act

perimeter = 2 \* (new\_height + new\_width);

//assert

Assert.AreEqual(perimeter, newRectangle.GetPerimeter());

}

[Test]

public void TestGetArea\_3()

{

//arrange

int area;

int new\_width = 76;

int new\_height = 13;

Rectangle newRectangle = new Rectangle(new\_width, new\_height);

//act

area = new\_width \* new\_height;

//assert

Assert.AreEqual(area, newRectangle.GetArea());

}

}

}