Create Aplicación de consola

To use the class go to Soltuion > Agregar > Proyecto existente and select Point2D > Point2D.csproj

Then add a reference in ConsoleApp to Point2D

Add to program.cs: using TPP.Lab04;

This means that: IncrementTwoDouble(ref x, ref y,1.5,2.0)

The hanges to the actual parameters are visible from the point of the method

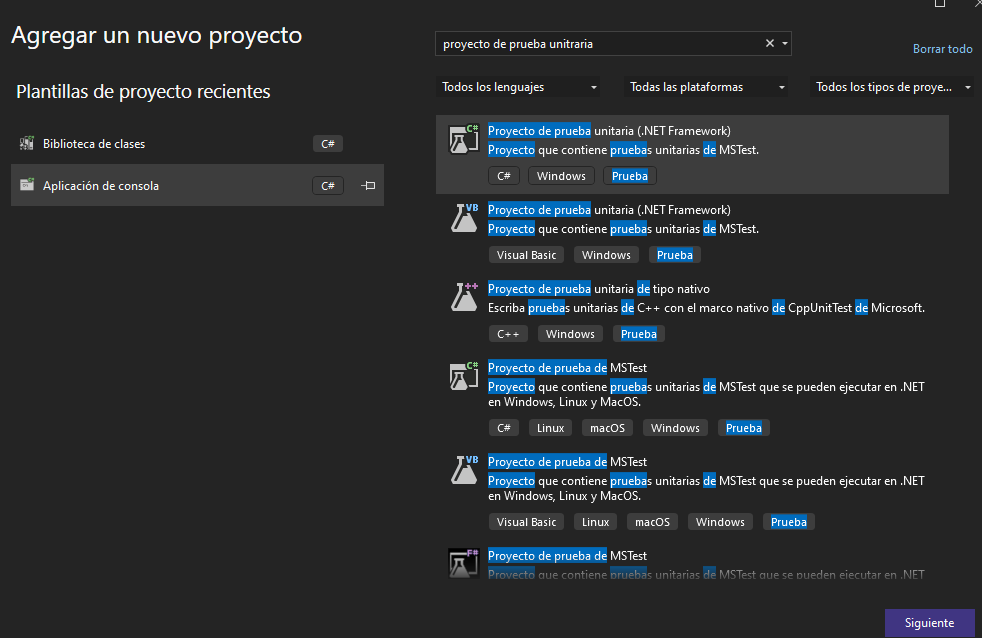
Parameter with out need to be modified in te body of the function. Otherwise you get an error. It’s aim is to return in the actual parameter a value

When you code an extension class:

Class mut be static

Method must be static

Parameters needs this keyword



Add the provided . Cs file, and copy the text

Prubea > Ejecutar todas las pruebas

O Prueba > Explorador de pruebas

namespace ComplexNumbers.Test

{

[TestClass]

public class TestComplejos

{

[TestMethod]

public void ModuleTest()

{

Complex a = new Complex(1.0, 1.0);

Assert.AreEqual(Math.Sqrt(2.0), a.Module()); //we can have rounding errors

Assert.AreEqual(Math.Abs(Math.Sqrt(2.0) - a.Module()) < 0.0001, true);

}

[TestMethod]

public void ConjugateTest()

{

Complex a = new Complex(1.0, 1.0),

valorEsperado = new Complex(1.0, -1.0),

valorObtenido;

valorObtenido = a.Conjugate();

Assert.AreEqual(valorEsperado.R, a.Conjugate().R);

Assert.AreEqual(valorEsperado.I, a.Conjugate().I);

}

[TestMethod]

public void AdditionTest()

{

Complex a = new Complex(1.0, 1.0),

b = new Complex(2.0, 3.0),

desiredValue = new Complex(3.0, 4.0),

obtainedValue;

obtainedValue = a + b;

Assert.AreEqual(desiredValue.R, obtainedValue.R);

Assert.AreEqual(desiredValue.I, obtainedValue.I);

}

//Add a test for \* representing two complex numbers product

[TestMethod]

public void ProductTest()

{

Complex a = new Complex(1.0, 1.0),

b = new Complex(2.0, 3.0),

desiredValue = new Complex(-1.0, 5.0),

obtainedValue;

obtainedValue = a \* b;

Assert.AreEqual(desiredValue.R, obtainedValue.R);

Assert.AreEqual(desiredValue.I, obtainedValue.I);

}

}

}