

EXAMEN PARCIAL - Calidad de software

CICLO 2023-2

ALUMNO: MANUEL ADEMAR FATAMA RUIZ CÓDIGO: 20192148F	Nota:
--	-------

Pregunta 1.- Programe en un desarrollo rojo-verde una clase Geom que tiene un método:

int **circleInside**(float x, float y, float radius, float centerX, float centerY)

Que determina si un punto X, Y se encuentra dentro de la circunferencia con centro en: (centerX, centerY) y radio radius.

La respuesta será:

- 1** – X, Y se encuentra dentro del círculo
- 0** – X, Y se encuentra sobre la circunferencia
- 1** – X, Y se encuentra fuera

Código de pruebas:

```
namespace RGRGeomTest
{
    [TestClass]
    public class GeomTestOriginCenter
    {
        private Geom g;
        private float x, y, radius, centerX, centerY;
        private int result;
        [TestInitialize]
        public void Init()
        {
            g = new Geom();
            radius = 4;
            centerX = 0;
            centerY = 0;
        }
        [TestMethod]
        public void TestInsideCircleRadius4()
        {
            // Arrange
            x = 2; y = 2;
            // Act
            result = g.CircleInside(x,y,radius,centerX,centerY);
            // Assert
            Assert.AreEqual(1,result);
        }
        [TestMethod]
```

```

    public void TestOutsideCircleRadius4()
    {
        // Arrange
        x = (float)-3.9; y = (float)2.0;
        // Act
        result = g.CircleInside(x, y, radius, centerX, centerY);
        // Assert
        Assert.AreEqual(-1, result);
    }

    [TestMethod]
    public void TestInCircunferenceRadius4()
    {
        // Arrange
        x = (float)Math.Sin(0.5)*radius; y = -(float)Math.Cos(0.5)*radius;

        // Act
        result = g.CircleInside(x, y, radius, centerX, centerY);
        // Assert
        Assert.AreEqual(0, result);
    }
}

```

Primera implementación:

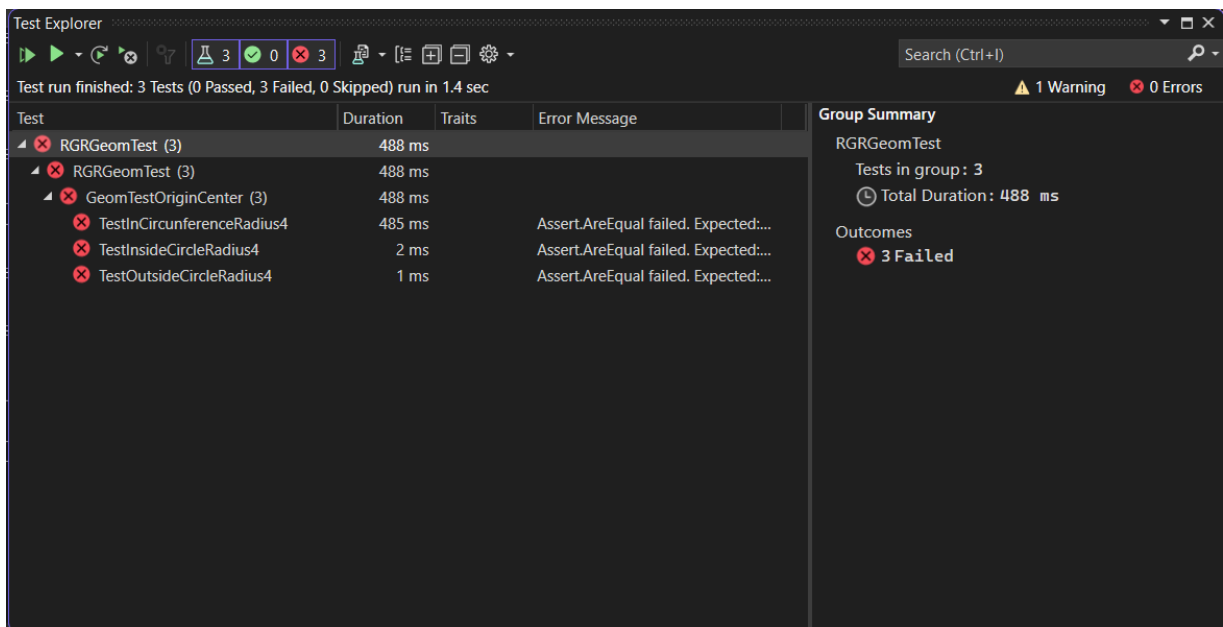
```

namespace RGRGeom
{
    internal class Program
    {
        static void Main(string[] args)
        {
        }
    }

    public class Geom
    {
        public int CircleInside(float x, float y, float radius, float centerX,
float centerY)
        {
            return -2;
        }
    }
}

```

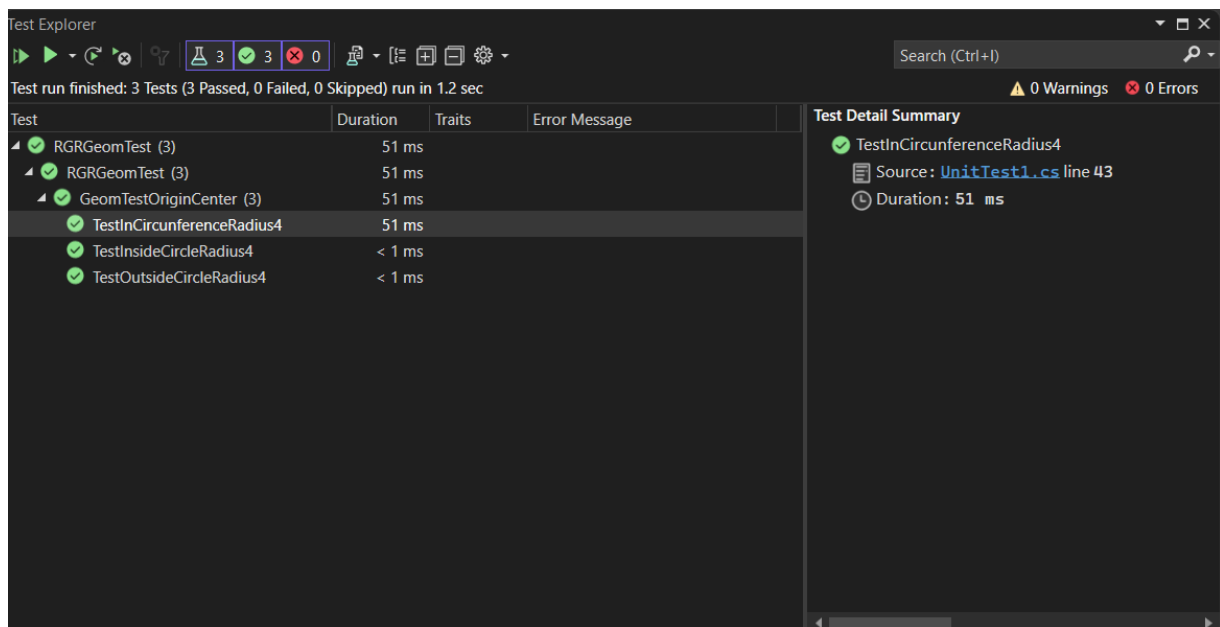
R:



G:

Corrección/Segunda implemetación

```
public class Geom
{
    public int CircleInside(float x, float y, float radius, float centerX,
float centerY)
    {
        var dist = (float)Math.Sqrt(Math.Pow(x - centerX, 2) + Math.Pow(y -
centerY, 2));
        if(radius > dist)
            return 1;
        if(radius < dist)
            return -1;
        if (Math.Abs(radius - dist)<0.1)
            return 0;
        return 2;
    }
}
```



Pregunta 2.- Programe una clase que haga un mocking de un carrito de compras, al cual se le puede agregar, quitar productos y hacer checkout.

Programe las pruebas de este carrito.

En este caso, usaré MST con .NET Framework 4.8 y Moq

```
public interface IProducto
{
    string Nombre { get; set; }
    decimal Precio { get; set; }
}

public class CarritoDeCompras
{
    private List<IProducto> productos;

    public CarritoDeCompras()
    {
        productos = new List<IProducto>();
    }

    public void AgregarProducto(IProducto producto)
    {
        productos.Add(producto);
    }

    public void QuitarProducto(IProducto producto)
    {
        productos.Remove(producto);
    }

    public decimal Checkout()
```

```

{
    decimal total = 0;
    foreach (var producto in productos)
    {
        total += producto.Precio;
    }
    // El carrito debe estar vacío después del checkout
    productos.Clear();
    return total;
}
}

```

Podemos apreciar que `IProducto` es una interfaz para el producto, pues estos pueden ser de diferentes tipos por lo que aprovecharían muy bien el polimorfismo. También se podría trabajar como una clase abstracta con `virtual` property (para que `Moq` pueda trabajar con ellas) pero se priorizó este enfoque para ahorrar tiempo.

El carrito de compras es una lista que contiene diferentes productos. Agrega y quita productos del carrito, además, luego de hacer el checkout este se tiene que vaciar.

Código de tests:

```

[TestClass]
public class TestCarritoCompras
{
    private CarritoDeCompras carrito;
    private Mock<IProducto> productoMock;

    [TestInitialize]
    public void TestInitialize()
    {
        carrito = new CarritoDeCompras();
        productoMock = new Mock<IProducto>();
    }

    [TestMethod]
    public void TestAgregarProducto()
    {
        // Arrange
        productoMock.Setup(p => p.Precio).Returns(100);
        // Act
        carrito.AgregarProducto(productoMock.Object);
        // Assert
        Assert.AreEqual(100, carrito.Checkout());
    }

    [TestMethod]
    public void TestQuitarProducto()
    {
        // Arrange
    }
}

```

```

        productoMock.Setup(p => p.Precio).Returns(60);
        //Act
        carrito.AgregarProducto(productoMock.Object);
        carrito.QuitarProducto(productoMock.Object);
        //Assert
        Assert.AreEqual(0, carrito.Checkout());
    }

    [TestMethod]
    public void TestCheckout()
    {
        var productoMock1 = new Mock<IProducto>();
        productoMock1.Setup(p => p.Precio).Returns(50);
        carrito.AgregarProducto(productoMock1.Object);

        var productoMock2 = new Mock<IProducto>();
        productoMock2.Setup(p => p.Precio).Returns(20);
        carrito.AgregarProducto(productoMock2.Object);

        Assert.AreEqual(70, carrito.Checkout());
        Assert.AreEqual(0, carrito.Checkout());
    }
}

```

Test Explorer

Test run finished: 3 Tests (3 Passed, 0 Failed, 0 Skipped) run in 3.4 sec

Test	Duration	Traits	Error Message
CarritoComprasMoqTest (3)	1.3 sec		
CarritoComprasMoqTest (3)	1.3 sec		
TestCarritoCompras (3)	1.3 sec		
TestAgregarProducto	1.3 sec		
TestCheckout	4 ms		
TestQuitarProducto	21 ms		

Group Summary

CarritoComprasMoqTest

Tests in group: 3

Total Duration: 1.3 sec

Outcomes

3 Passed

Pregunta 3.- Explique qué es un Marco de calidad y cuáles son los principales Marcos de calidad para el software.

Son una serie de reglas/limitaciones que se utiliza para calificar la calidad de un producto pues la calidad es algo subjetivo y entre sus interpretaciones está la de adecuación al uso, adaptación al

usuario, etc. Llamémosle delimitar u orientar hacia donde o cómo se va a “medir” la calidad de un producto.

Un ejemplo de marcos de calidad de software podría ser los niveles de acción en la ing. De software: Pues se realiza a nivel individual (como desarrollador, por ejemplo) y es ahí cuando cumples algunas reglas de diseño tuyas, pero también estás sometido a las del cliente y de empresa. Está también a nivel de empresa, es aquí donde se centra en establecer una cultura de calidad en toda la organización, donde se siguen otro tipo de “reglas” o marcos como puede ser el ISO 25010:2011 por mencionar uno. Y también a nivel de proyecto, pues también cabe recalcar que diferentes productos no pueden ser tratados de la misma forma ni mucho menos comparados en sus requisitos ni otros aspectos que son importantes para “medir” su calidad.

Pregunta 4.- Escriba las historias de uso para Ud. como estudiante matriculado en el curso de calidad de software y el proceso de rendir examen y consultar su evaluación

ID	Enunciado de la historia				Criterios de aceptación			
	Rol	Característica/ Funcionalidad	Razón/ Resultado	Nro. De escenario	Criterio de Aceptacion (titulo)	Contexto	Evento	Resultado/ Comportamiento esperado
1	Como estudiante matriculado	Quiero acceder a la actividad de examen	Para poder rendir el examen	1	Acceder a actividad publicada	Al ingresar al aula virtual del curso en el horario de examen	Se revisa que la actividad examen que está colgada	Se muestre el examen con sus pautas (tiempo, indicaciones, preguntas, archivos a usar)
2	Como estudiante matriculado	Quiero enviar mi resolución del examen	Para poder ser calificado por el docente	1	Entregar archivos de solución a tiempo	Al ingresar a la actividad de examen y habiendo terminado de resolverlo en el tiempo estipulado	Se suben los archivos que van a ser calificados	Se muestra el estado de los archivos como enviados y se registra envío como envío temprano
				2	Entregar archivos de solución fuera de tiempo	Al ingresar a la actividad de examen y habiendo terminado de resolverlo		Se muestra el estado de los archivos como enviados y se registra envío tardío
3	Como estudiante matriculado	Quiero consultar mi calificación del examen	Para conocer mi rendimiento	1	Calificación no realizada	Al ingresar a la actividad de examen	El docente aún no corrige el examen	Se muestra el estado de calificación como pendiente
				2	Calificación realizada		El docente corrigió el examen	Se muestra la calificación otorgada por el docente