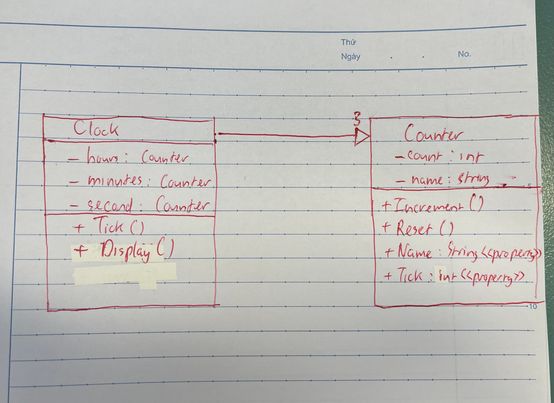
**3.1P**

**UML Diagram**

****

**Program file**

namespace \_3.\_1

{

internal class Program

{

static void Main(string[] args)

{

Clock clock = new Clock();

for (int i = 0; i < 86400; i++)

{

clock.Tick();

clock.Display();

}

}

}

}

**Counter class**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_3.\_1

{

public class Counter

{

private int \_count;

private string \_name;

public Counter(string name)

{

\_name = name;

\_count = 0;

}

public void Increment()

{

\_count++;

}

public void Reset()

{

\_count = 0;

}

public string Name

{

get { return \_name; }

set { \_name = value; }

}

public int Ticks

{

get { return \_count; }

}

}

}

**Clock class**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_3.\_1

{

public class Clock

{

private Counter \_hours;

private Counter \_minutes;

private Counter \_seconds;

public Clock()

{

\_hours = new Counter("Hours");

\_minutes = new Counter("Minutes");

\_seconds = new Counter("Seconds");

}

public void Tick()

{

\_seconds.Increment();

if (\_seconds.Ticks == 60)

{

\_seconds.Reset();

\_minutes.Increment();

if (\_minutes.Ticks == 60)

{

\_minutes.Reset();

\_hours.Increment();

if (\_hours.Ticks == 24)

{

\_hours.Reset();

}

}

}

}

public void Reset()

{

\_hours.Reset();

\_minutes.Reset();

\_seconds.Reset();

}

public void Display()

{

Console.WriteLine($"{ \_hours.Ticks:D2}: {\_minutes.Ticks:D2}: {\_seconds.Ticks:D2}");

}

}

}

**Output**

**A screenshot of a computer

Description automatically generated**

**The clock runs till it hits 0:0:0 then stops**

**Test File**

using \_3.\_1;

namespace TestProject2

{

public class Tests

{

[SetUp]

public void Setup()

{

}

Clock clock = new Clock();

[Test]

public void Test1()

{

clock.Reset();

Assert.That(clock.Display, Is.EqualTo("00: 00: 00"));

}

[Test]

public void Test2()

{

clock.Reset();

clock.Tick();

Assert.That(clock.Display, Is.EqualTo("00: 00: 01"));

}

[Test]

public void Test3()

{

clock.Reset();

for (int i = 0; i < 60; i++)

{

clock.Tick();

}

Assert.That(clock.Display, Is.EqualTo("00: 01: 00"));

}

[Test]

public void Test4()

{

clock.Reset();

for (int i = 0; i < 3600; i++)

{

clock.Tick();

}

Assert.That(clock.Display, Is.EqualTo("01: 00: 00"));

}

[Test]

public void Test5()

{

clock.Reset();

for (int i = 0; i < 86400; i++)

{

clock.Tick();

}

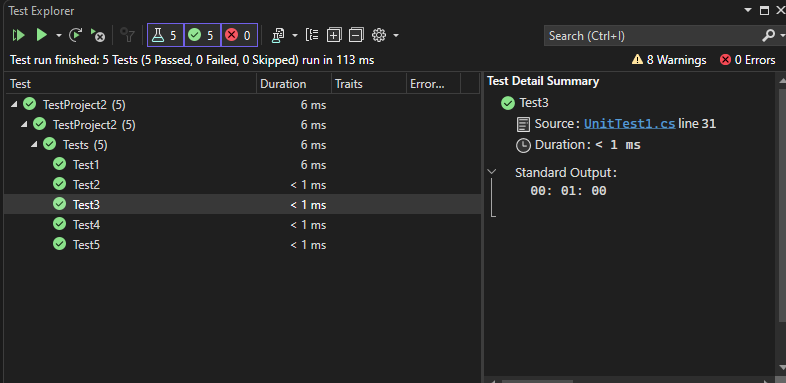
Assert.That(clock.Display, Is.EqualTo("00: 00: 00"));

}

}

}

**Test Result**

****