Selected files

5 printable files

Week 3\3.1\Clock\Clock.cs
Week 3\3.1\Clock\Counter.cs
Week 3\3.1\Clock\Program.cs
Week 3\3.1\ClockTest\ClockTest.cs
Week 3\3.1\ClockTest\CounterTest.cs

Week 3\3.1\Clock\Clock.cs

```
1  using System;
2
3  namespace ClockProgram
4  {
5      public class Clock
6      {
7          private Counter _seconds;
8          private Counter _minutes;
9          private Counter _hours;
10
11         public Clock()
12         {
13             _seconds = new Counter("Seconds");
14             _minutes = new Counter("Minutes");
15             _hours = new Counter("Hours");
16         }
17
18         public void Tick()
19         {
20             _seconds.Increment();
21             if (_seconds.Ticks != 60) return;
22
23             _seconds.Reset();
24             _minutes.Increment();
25             if (_minutes.Ticks != 60) return;
26
27             _minutes.Reset();
```

```
            _hours.Increment();
            if (_hours.Ticks != 12) return;

            _hours.Reset();
        }

        public void Reset()
        {
            _seconds.Reset();
            _minutes.Reset();
            _hours.Reset();
        }

        public string Time
        {
            get
            {
                return $"{_hours.Ticks:D2}:{_minutes.Ticks:D2}:{_seconds.Ticks:D2}";
            }
        }
    }
}
```

Week 3\3.1\Clock\Counter.cs

```
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;
        }
    }
}
```

Week 3\3.1\Clock\Program.cs

```
namespace ClockProgram
{
    public static class Program
    {
        public static void Main()
        {
            Clock clock = new Clock();

```

```
            for (int i = 0; i < 5; i++)
            {
                clock.Tick();
                System.Console.WriteLine(clock.Time);
            }
        }
    }
}
```

Week 3\3.1\ClockTest\ClockTest.cs

```
using NUnit.Framework;
using ClockProgram;

namespace TestClock
{
    [TestFixture]
    public class ClockTest
    {
        [TestCase]
        public void TestInitializeClock()
        {
            Clock clock = new Clock();
            Assert.That(clock.Time, Is.EqualTo("00:00:00"));
        }

        [TestCase]
        public void TestTick()
        {
            Clock clock = new Clock();
            clock.Tick();
            Assert.That(clock.Time, Is.EqualTo("00:00:01"));
        }

        [Test]
        public void TestTickMultiple()
        {
            Clock clock = new Clock();
            for (int i = 0; i < 3; i++)
```

```
            {
                clock.Tick();
            }
            Assert.That(clock.Time, Is.EqualTo("00:00:03"));
        }

        [Test]
        public void TestTickMinutes()
        {
            Clock clock = new Clock();
            for (int i = 0; i < 60; i++)
            {
                clock.Tick();
            }
            Assert.That(clock.Time, Is.EqualTo("00:01:00"));
        }

        [Test]
        public void TestTickHours()
        {
            Clock clock = new Clock();
            for (int i = 0; i < 3600; i++)
            {
                clock.Tick();
            }
            Assert.That(clock.Time, Is.EqualTo("01:00:00"));
        }

        [Test]
        public void TestTickHalfDay()
        {
            Clock clock = new Clock();
            for (int i = 0; i < 43200; i++)
            {
                clock.Tick();
            }
            Assert.That(clock.Time, Is.EqualTo("00:00:00"));
        }
```

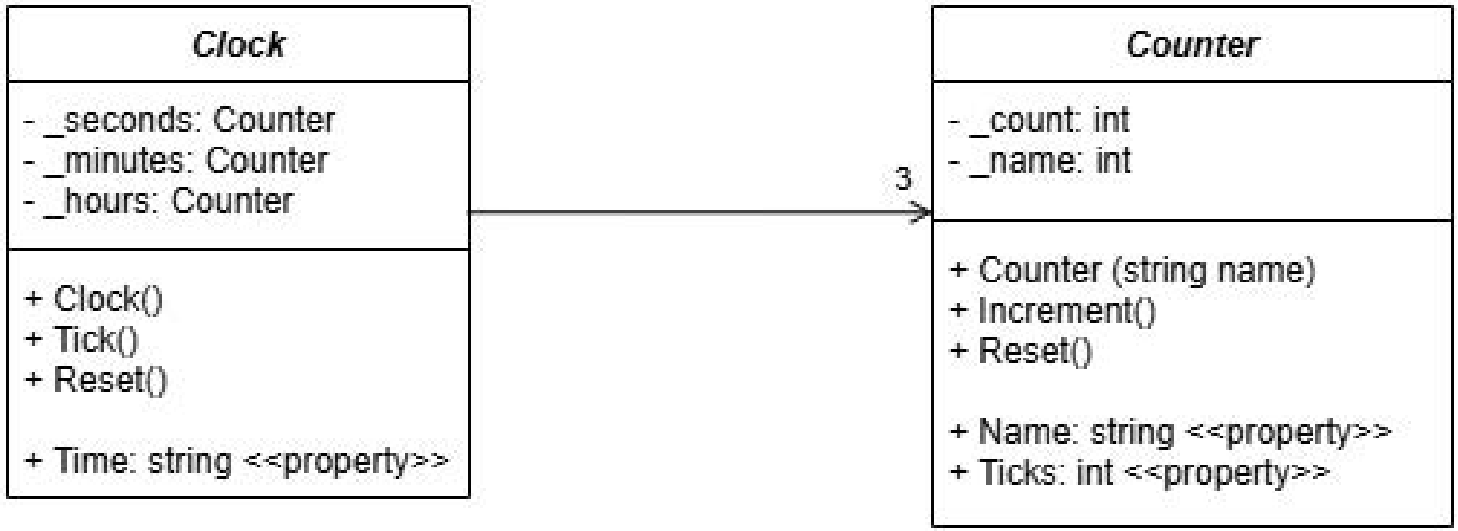
```
    }
}
```

Week 3\3.1\ClockTest\CounterTest.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using NUnit.Framework;

namespace TestCounter
{
    [TestFixture]
    public class CounterTest
    {
        [Test]
        public void TestInitializeCounter()
        {
            Counter a = new Counter("A");
            Assert.That(a.Ticks, Is.EqualTo(0));
        }

        [Test]
        public void TestIncrement()
        {
            Counter a = new Counter("A");
            a.Increment();
            Assert.That(a.Ticks, Is.EqualTo(1));
        }

        [Test]
        public void TestIncrementMultiple()
        {
            Counter a = new Counter("A");
            for (int i = 0; i < 3; i++)
            {
                a.Increment();
            }
```

```
35              Assert.That(a.Ticks, Is.EqualTo(3));
36          }
37
38          [Test]
39          public void TestReset()
40          {
41              Counter a = new Counter("A");
42              a.Increment();
43              a.Reset();
44              Assert.That(a.Ticks, Is.EqualTo(0));
45          }
46      }
47  }
```

UML Diagram:



Screen shot of output and unit test: