

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();
```

```

28         _hours.Increment();
29         if (_hours.Ticks != 12) return;
30
31         _hours.Reset();
32     }
33
34     public void Reset()
35     {
36         _seconds.Reset();
37         _minutes.Reset();
38         _hours.Reset();
39     }
40
41     public string Time
42     {
43         get
44         {
45             return $"{_hours.Ticks:D2}:{_minutes.Ticks:D2}:{_seconds.Ticks:D2}";
46         }
47     }
48 }
49 }

```

Week 3\3.1\Clock\Counter.cs

```

1  public class Counter
2  {
3      private int _count;
4      private string _name;
5
6      public Counter(string name)
7      {
8          _name = name;
9          _count = 0;
10     }
11
12     public void Increment()
13     {
14         _count++;

```

```

15     }
16
17     public void Reset()
18     {
19         _count = 0;
20     }
21
22     public string Name
23     {
24         get
25         {
26             return _name;
27         }
28
29         set
30         {
31             _name = value;
32         }
33     }
34
35     public int Ticks
36     {
37         get
38         {
39             return _count;
40         }
41     }
42 }

```

Week 3\3.1\Clock\Program.cs

```

1 namespace ClockProgram
2 {
3     public static class Program
4     {
5         public static void Main()
6         {
7             Clock clock = new Clock();
8

```

```

9         for (int i = 0; i < 5; i++)
10        {
11            clock.Tick();
12            System.Console.WriteLine(clock.Time);
13        }
14    }
15 }
16 }

```

Week 3\3.1\ClockTest\ClockTest.cs

```

1  using NUnit.Framework;
2  using ClockProgram;
3
4  namespace TestClock
5  {
6      [TestFixture]
7      public class ClockTest
8      {
9          [TestCase]
10         public void TestInitializeClock()
11         {
12             Clock clock = new Clock();
13             Assert.That(clock.Time, Is.EqualTo("00:00:00"));
14         }
15
16         [TestCase]
17         public void TestTick()
18         {
19             Clock clock = new Clock();
20             clock.Tick();
21             Assert.That(clock.Time, Is.EqualTo("00:00:01"));
22         }
23
24         [Test]
25         public void TestTickMultiple()
26         {
27             Clock clock = new Clock();
28             for (int i = 0; i < 3; i++)

```

```
29     {
30         clock.Tick();
31     }
32     Assert.That(clock.Time, Is.EqualTo("00:00:03"));
33 }
34
35 [Test]
36 public void TestTickMinutes()
37 {
38     Clock clock = new Clock();
39     for (int i = 0; i < 60; i++)
40     {
41         clock.Tick();
42     }
43     Assert.That(clock.Time, Is.EqualTo("00:01:00"));
44 }
45
46 [Test]
47 public void TestTickHours()
48 {
49     Clock clock = new Clock();
50     for (int i = 0; i < 3600; i++)
51     {
52         clock.Tick();
53     }
54     Assert.That(clock.Time, Is.EqualTo("01:00:00"));
55 }
56
57 [Test]
58 public void TestTickHalfDay()
59 {
60     Clock clock = new Clock();
61     for (int i = 0; i < 43200; i++)
62     {
63         clock.Tick();
64     }
65     Assert.That(clock.Time, Is.EqualTo("00:00:00"));
66 }
```

```
67     }
68 }
```

Week 3\3.1\ClockTest\CounterTest.cs

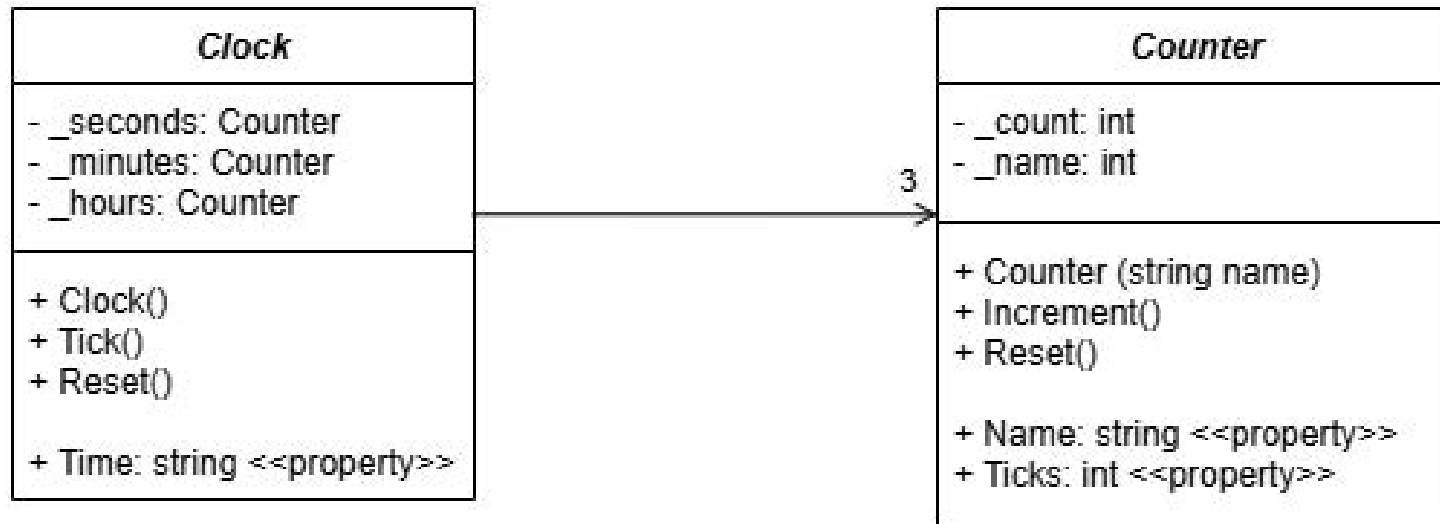
```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using NUnit.Framework;
6
7  namespace TestCounter
8  {
9      [TestFixture]
10     public class CounterTest
11     {
12         [Test]
13         public void TestInitializeCounter()
14         {
15             Counter a = new Counter("A");
16             Assert.That(a.Ticks, Is.EqualTo(0));
17         }
18
19         [Test]
20         public void TestIncrement()
21         {
22             Counter a = new Counter("A");
23             a.Increment();
24             Assert.That(a.Ticks, Is.EqualTo(1));
25         }
26
27         [Test]
28         public void TestIncrementMultiple()
29         {
30             Counter a = new Counter("A");
31             for (int i = 0; i < 3; i++)
32             {
33                 a.Increment();
34             }
35         }
36     }
37 }
```

```

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:

The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the `Program.cs` file with the following code:

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

            for (int i = 0; i < 5; i++)
            {
            }
        }
    }
}
```
- Test Explorer:** Displays the test results for the `ClockTest` suite. The table below summarizes the tests:

Test	Duration	Traits	Error Message
ClockTest (10)	12 ms		
TestClock (6)	10 ms		
TestInitializeClock()	10 ms		
TestTick()	< 1 ms		
TestTickHalfDay	< 1 ms		
TestTickHours	< 1 ms		
TestTickMinutes	< 1 ms		
TestTickMultiple	< 1 ms		
TestCounter (4)	2 ms		
CounterTest (4)	2 ms		
TestIncrement	2 ms		
TestIncrementMultiple	< 1 ms		
TestInitializeCounter	< 1 ms		

- Group Summary:** Shows the overall test results:
 - Tests in group: 10
 - Total Duration: 12
 - Outcomes: 10 Passed
- Debug Console:** Displays the output of the application:

```
00:00:01
00:00:02
00:00:03
00:00:04
00:00:05

C:\Users\Admin\Desktop\COS20007-OOP\Week 3\3.1\Clock\bin\Debug\net8.0\Clock.exe (process 1552) exited with code 0.
Press any key to close this window . . .
```