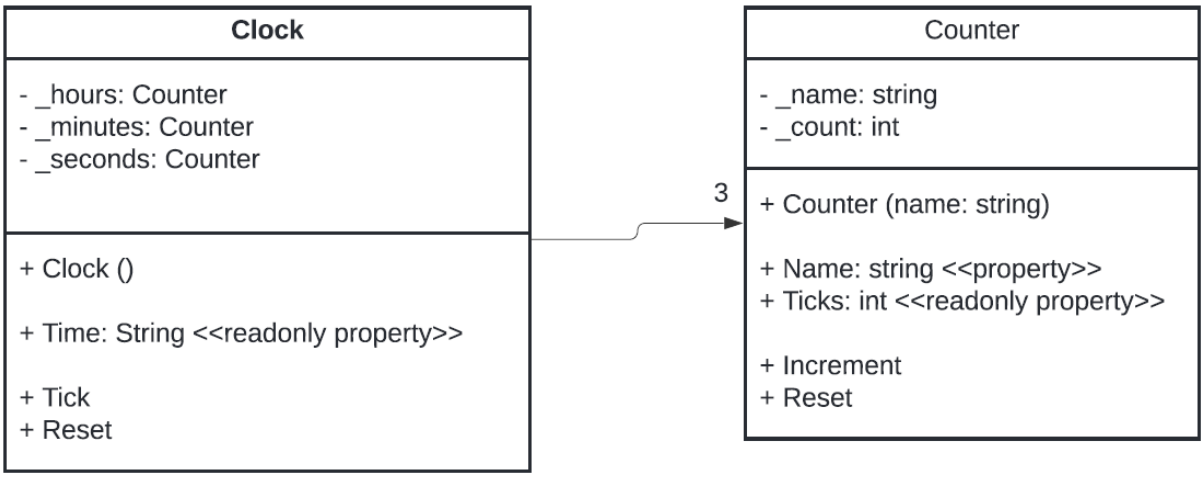
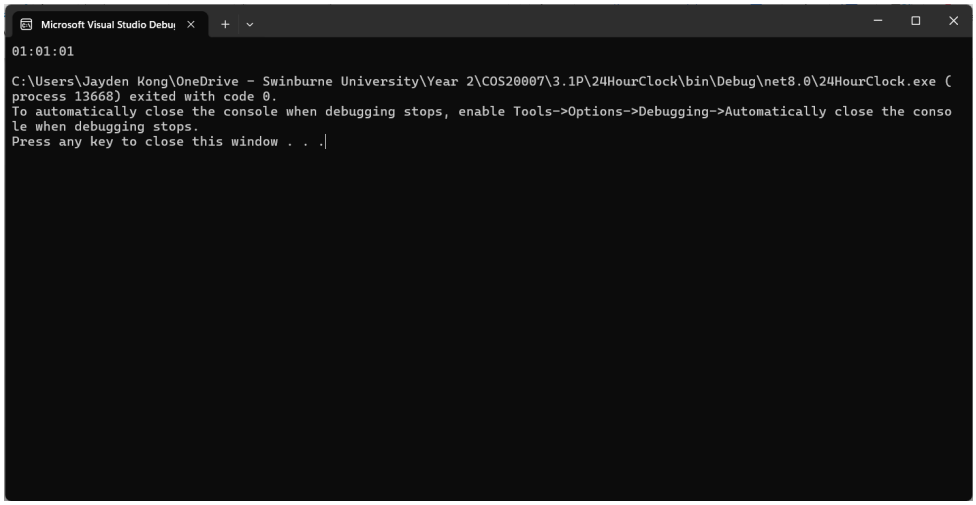
3.1P - Clock Class

Jayden Kong, 104547242

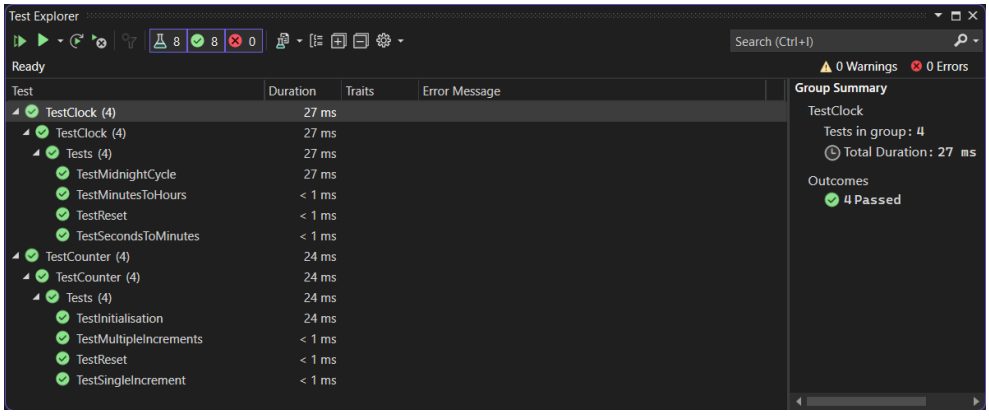
Class UML diagram:



Example of program output:



Unit tests passing:



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace _24HourClock
8  {
9      public class Clock
10     {
11         private Counter _seconds;
12         private Counter _minutes;
13         private Counter _hours;
14
15         public string Time
16         {
17             get
18             {
19                 return string.Format("{0:00}:{1:00}:{2:00}", _hours.Ticks,
                       _minutes.Ticks, _seconds.Ticks);
20             }
21         }
22
23         public Clock()
24         {
25             _seconds = new Counter("seconds");
26             _minutes = new Counter("minutes");
27             _hours = new Counter("hours");
28         }
29
30         public void Tick()
31         {
32             _seconds.Increment();
33             if (_seconds.Ticks == 60)
34             {
35                 _minutes.Increment();
36                 _seconds.Reset();
37             }
38             if (_minutes.Ticks == 60)
39             {
40                 _hours.Increment();
41                 _minutes.Reset();
42             }
43             if ( _hours.Ticks == 24)
44             {
45                 _hours.Reset();
46             }
47         }
48
```

```
49          public void Reset()
50          {
51              _seconds.Reset();
52              _minutes.Reset();
53              _hours.Reset();
54          }
55
56      }
57  }
58
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace _24HourClock
8  {
9      public class Counter
10     {
11         private int _count;
12         private string _name;
13         public string Name
14         {
15             get
16             {
17                 return _name;
18             }
19             set
20             {
21                 _name = value;
22             }
23         }
24
25         public int Ticks
26         {
27             get
28             {
29                 return _count;
30             }
31         }
32
33         public Counter(string name)
34         {
35             _name = name;
36             _count = 0;
37         }
38
39         public void Increment()
40         {
41             _count += 1;
42         }
43         public void Reset()
44         {
45             _count = 0;
46         }
47     }
48 }
49
```

```
1  namespace _24HourClock
2  {
3      internal class Program
4      {
5          static void Main(string[] args)
6          {
7              Clock myClock = new Clock();
8              for (int i = 0; i < 3661; i++)
9              {
10                 myClock.Tick();                 // Advances the clock
                     forward by 3661 seconds
11             }
12             Console.WriteLine(myClock.Time);    // Should read "01:01:01"
13         }
14     }
15 }
16
```

```
1  using _24HourClock;
2
3  namespace TestCounter
4  {
5      public class Tests
6      {
7
8          [Test]
9          public void TestInitialisation()
10         {
11             Counter testCounter = new Counter("test");
12             int value = testCounter.Ticks;
13             Assert.That(value, Is.EqualTo(0));
14         }
15
16         [Test]
17         public void TestSingleIncrement()
18         {
19             Counter testCounter = new Counter("test");
20             testCounter.Increment();
21             int value = testCounter.Ticks;
22             Assert.That(value, Is.EqualTo(1));
23         }
24
25         [Test]
26         public void TestMultipleIncrements()
27         {
28             Counter testCounter = new Counter("test");
29             for (int i = 0; i < 10; i++)
30             {
31                 testCounter.Increment();
32
33             }
34             int value = testCounter.Ticks;
35             Assert.That(value, Is.EqualTo(10));
36         }
37
38         [Test]
39         public void TestReset()
40         {
41             Counter testCounter = new Counter("test");
42             for (int i = 0; i < 10; i++)
43             {
44                 testCounter.Increment();
45
46             }
47             testCounter.Reset();
48             int value = testCounter.Ticks;
49             Assert.That(value, Is.EqualTo(0));
```

```
50          }
51      }
52  }
```

```
1  using _24HourClock;
2
3  namespace TestClock
4  {
5      public class Tests
6      {
7
8          [Test]
9          public void TestSecondsToMinutes()
10         {
11             // testing for 60 seconds
12             Clock testClock = new Clock();
13             for (int i = 0; i < 60; i++)
14             {
15                 testClock.Tick();
16             }
17             Assert.That(testClock.Time, Is.EqualTo("00:01:00"));
18         }
19
20         [Test]
21         public void TestMinutesToHours()
22         {
23             // testing for 60 minutes
24             Clock testClock = new Clock();
25             for (int i = 0; i < 3600; i++)
26             {
27                 testClock.Tick();
28             }
29             Assert.That(testClock.Time, Is.EqualTo("01:00:00"));
30         }
31
32         [Test]
33         public void TestMidnightCycle()
34         {
35             // testing for 24:00:00 changing to 00:00:00
36             Clock testClock = new Clock();
37             for (int i = 0; i < 86400; i++)
38             {
39                 testClock.Tick();
40             }
41             Assert.That(testClock.Time, Is.EqualTo("00:00:00"));
42         }
43
44         [Test]
45         public void TestReset()
46         {
47             Clock testClock = new Clock();
48             for (int i = 0; i < 8642; i++)
49             {
```

```
50                  testClock.Tick();
51              }
52              testClock.Reset();
53              Assert.That(testClock.Time, Is.EqualTo("00:00:00"));
54          }
55
56      }
57  }
```