

SWINBURNE UNIVERSITY OF TECHNOLOGY

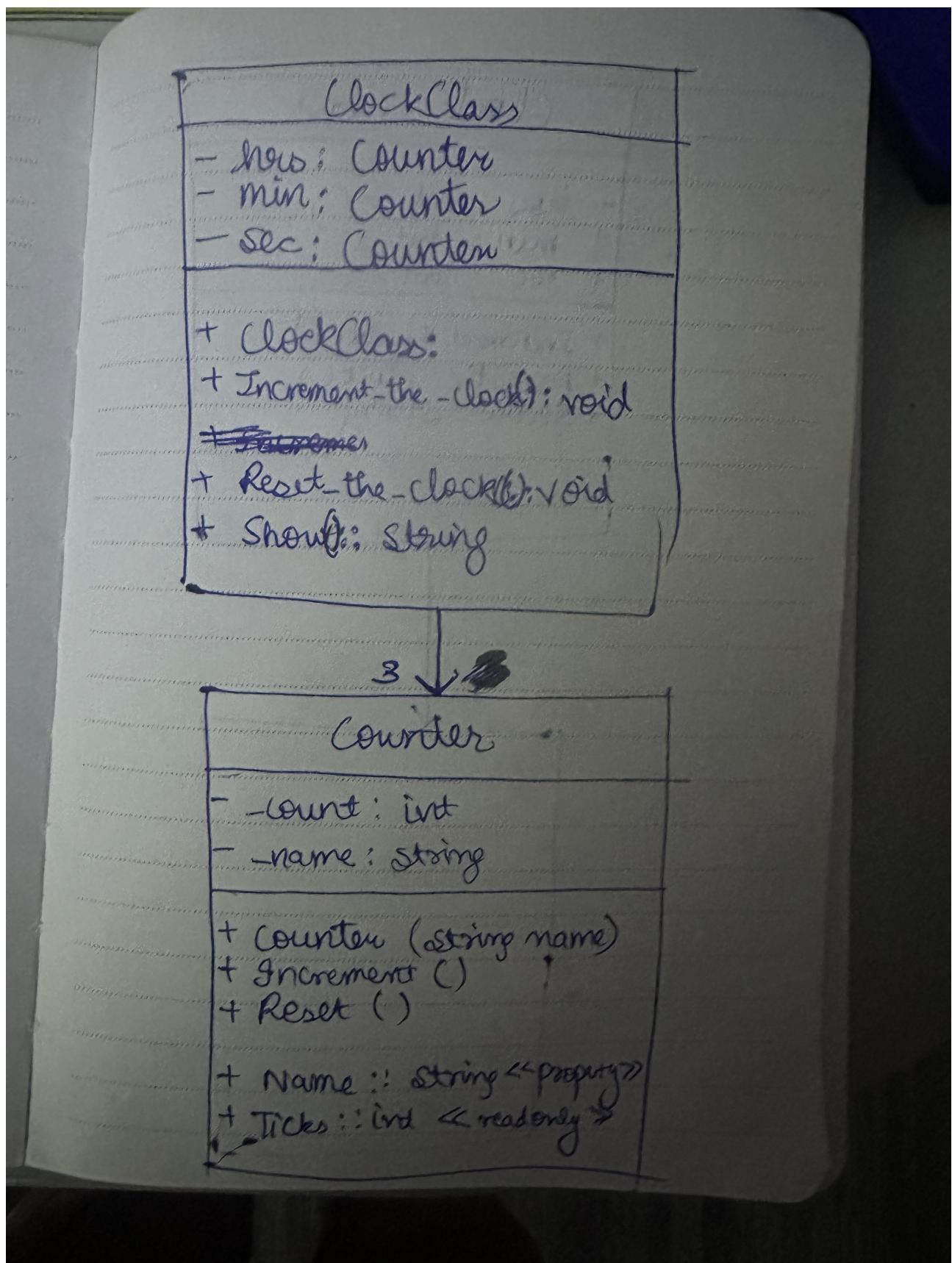
COS20007 OBJECT ORIENTED PROGRAMMING

---

## Clock Class

---

PDF generated at 19:14 on Monday 28<sup>th</sup> August, 2023



```
1  using System;
2  using NUnit;
3  namespace Task3._1_Clock
4  {
5      public class Program
6      {
7          public static void Main()
8          {
9              ClockClass clk = new ClockClass();
10
11             for (int i = 0; i < 86400; i = i + 1)
12             {
13                 clk.Increment_the_clock();
14                 Console.WriteLine(clk.Show());
15
16             }
17             Console.ReadLine();
18
19         }
20
21     }
22 }
23
```

```
1  using System;
2  namespace Task3._1_Clock
3  {
4      public class ClockClass
5      {
6          Counter hrs = new Counter ("hrs");
7          Counter min = new Counter("min");
8          Counter sec = new Counter("sec");
9
10         public ClockClass()
11         {
12             //int hrs = 0;
13             //int min = 0;
14
15             //int sec = 0;
16
17         }
18         public void Increment_the_clock()
19         {
20             sec.Increment();
21             if (sec.Ticks == 60)
22             {
23                 sec.Reset();
24                 min.Increment();
25
26                 if (min.Ticks == 60 )
27                 {
28                     min.Reset();
29                     hrs.Increment();
30
31                     if (hrs.Ticks == 24)
32                     {
33                         hrs.Reset();
34                     }
35                 }
36             }
37
38         }
39     }
40
41     public void Reset_the_clock()
42     {
43         hrs.Reset(); min.Reset(); sec.Reset();
44     }
45
46     public string Show()
47     {
48         return (hrs.Ticks.ToString("00") + " : " + min.Ticks.ToString("00") + " :
49             " + sec.Ticks.ToString("00"));
50     }
51 }
52 }
```



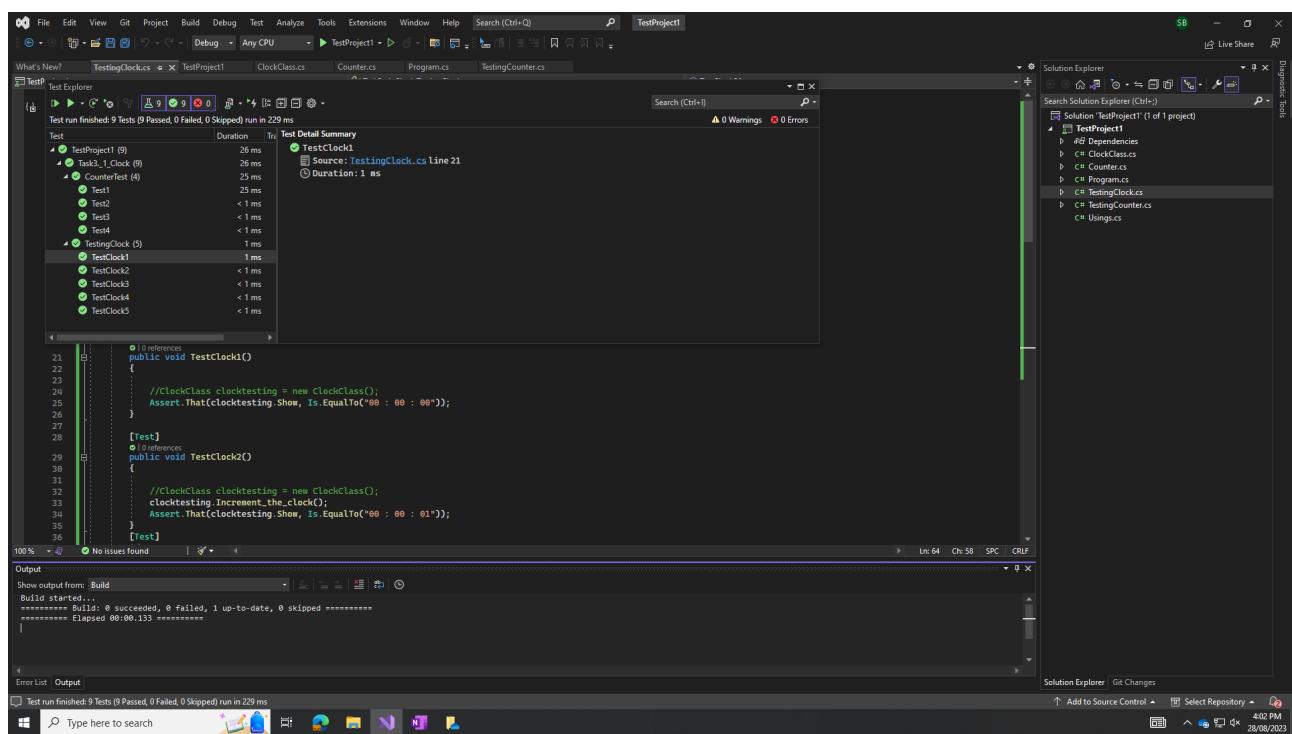
```
1  using NUnit;
2  using NUnit.Framework;
3  using Task3._1_Clock;
4  namespace Task3._1_Clock
5  {
6      [TestFixture]
7      public class TestingClock
8      {
9          ClockClass clocktesting;
10         public void TestClock()
11         {
12
13         }
14         [SetUp]
15         public void Setup()
16         {
17             clocktesting = new ClockClass();
18         }
19         [Test]
20         public void TestClock1()
21         {
22
23             //ClockClass clocktesting = new ClockClass();
24             Assert.That(clocktesting.Show, Is.EqualTo("00 : 00 : 00"));
25         }
26
27         [Test]
28         public void TestClock2()
29         {
30
31             //ClockClass clocktesting = new ClockClass();
32             clocktesting.Increment_the_clock();
33             Assert.That(clocktesting.Show, Is.EqualTo("00 : 00 : 01"));
34         }
35         [Test]
36         public void TestClock3()
37         {
38             //ClockClass clocktesting = new ClockClass();
39             clocktesting.Increment_the_clock();
40             clocktesting.Reset_the_clock();
41             Assert.That(clocktesting.Show, Is.EqualTo("00 : 00 : 00"));
42         }
43         [Test]
44         public void TestClock4()
45         {
46             for (int a = 0; a < 60; a++)
47             {
48                 clocktesting.Increment_the_clock();
49
50             }
51             Assert.That(clocktesting.Show, Is.EqualTo("00 : 01 : 00"));
52
53 }
```

```
54
55    }
56    [Test]
57    public void TestClock5()
58    {
59        for (int a = 0; a < 3600; a++)
60        {
61            clocktesting.Increment_the_clock();
62        }
63        Assert.That(clocktesting.Show, Is.EqualTo("01 : 00 : 00"));
64
65
66    }
67}
68}
69}
70}
```

```
1  using System;
2  using Newtonsoft.Json.Linq;
3  using System.Xml.Linq;
4  using NUnit;
5
6
7  namespace Task3._1_Clock
8  {
9      public class Counter
10     {
11         private int _count;
12         private string _name;
13
14         public Counter(string name)
15         {
16             _name = name;
17             _count = 0;
18         }
19         public void Increment()
20         {
21             _count++;
22         }
23         public void Reset()
24         {
25             _count = 0;
26         }
27         public string Name
28         {
29             get
30             {
31                 return _name;
32             }
33             set
34             {
35                 _name = value;
36             }
37         }
38         public int Ticks
39         {
40             get
41             {
42                 return _count;
43             }
44         }
45     }
46 }
47 }
```

```
1  using NUnit;
2  using NUnit.Framework;
3  using Task3._1_Clock;
4  namespace Task3._1_Clock
5  {
6      [TestFixture()]
7      public class CounterTest
8      {
9          Counter countertesting;
10         public void TestCounterTest()
11         {
12
13     }
14     [SetUp]
15     public void Setup()
16     {
17         countertesting = new Counter("counter");
18     }
19
20     [Test]
21     public void Test1()
22     {
23         //Counter countertesting = new Counter("counter");
24         Assert.That(countertesting.Ticks, Is.EqualTo(0));
25         Assert.Pass();
26     }
27     [Test]
28     public void Test2()
29     {
30         //Counter countertesting = new Counter("counter");
31         countertesting.Increment();
32         int incrementedcounter = countertesting.Ticks;
33         Assert.That(incrementedcounter, Is.EqualTo(1));
34     }
35     [Test]
36     public void Test3()
37     {
38         //Counter countertesting = new Counter("counter");
39         countertesting.Increment();
40         countertesting.Increment();
41         countertesting.Increment();
42
43         int incrementedcounterthrice = countertesting.Ticks;
44         Assert.That(incrementedcounterthrice, Is.EqualTo(3));
45     }
46     [Test]
47     public void Test4()
48     {
49         //Counter countertesting = new Counter("counter");
50         countertesting.Increment();
51
52         countertesting.Reset();
53         int counterresetting = countertesting.Ticks;
```

```
54         Assert.That(counterresetting, Is.EqualTo(0));  
55     }  
56     }  
57 }  
58 }
```



The screenshot shows the Microsoft Visual Studio IDE interface during a diagnostic session. The main window displays the code for `TestingClock.cs` in the `TestProject1` solution. The code uses `Task3_1.Clock` and `Task3_1.ClockTesting` from the `Task3_1` namespace. The diagnostic tools window on the right shows a timeline from 16 seconds ago, with a chart for Process Memory (MB) and CPU (% of all processors) usage. The memory usage remains constant at 18 MB, while CPU usage is near zero. The diagnostic session summary indicates 10 seconds. The status bar at the bottom right shows the date and time as 28/08/2023 4:02 PM.

```
1  using NUnit;
2  using NUnit.Framework;
3  using Task3_1.Clock;
4  namespace Task3_1.Clock
5  {
6      [TestFixture]
7      public class TestingClock
8      {
9          [ClockClass]
10         public void Test()
11         {
12             var clock = new Clock();
13             clock.Increment();
14             Assert.That(clock.Show, Is.EqualTo("00 : 00 : 01"));
15         }
16         [SetUp]
17         public void Setup()
18         {
19             var clock = new Clock();
20             clock.Increment();
21             Assert.That(clock.Show, Is.EqualTo("00 : 00 : 00"));
22         }
23         [Test]
24         public void TestClock2()
25         {
26             var clock = new Clock();
27             clock.Increment();
28             Assert.That(clock.Show, Is.EqualTo("00 : 00 : 01"));
29         }
30     }
31 }
```