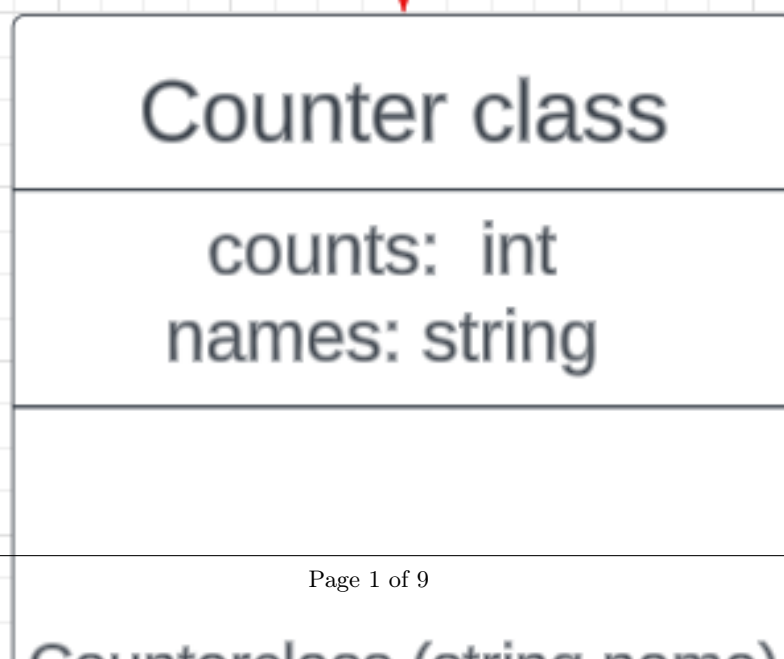
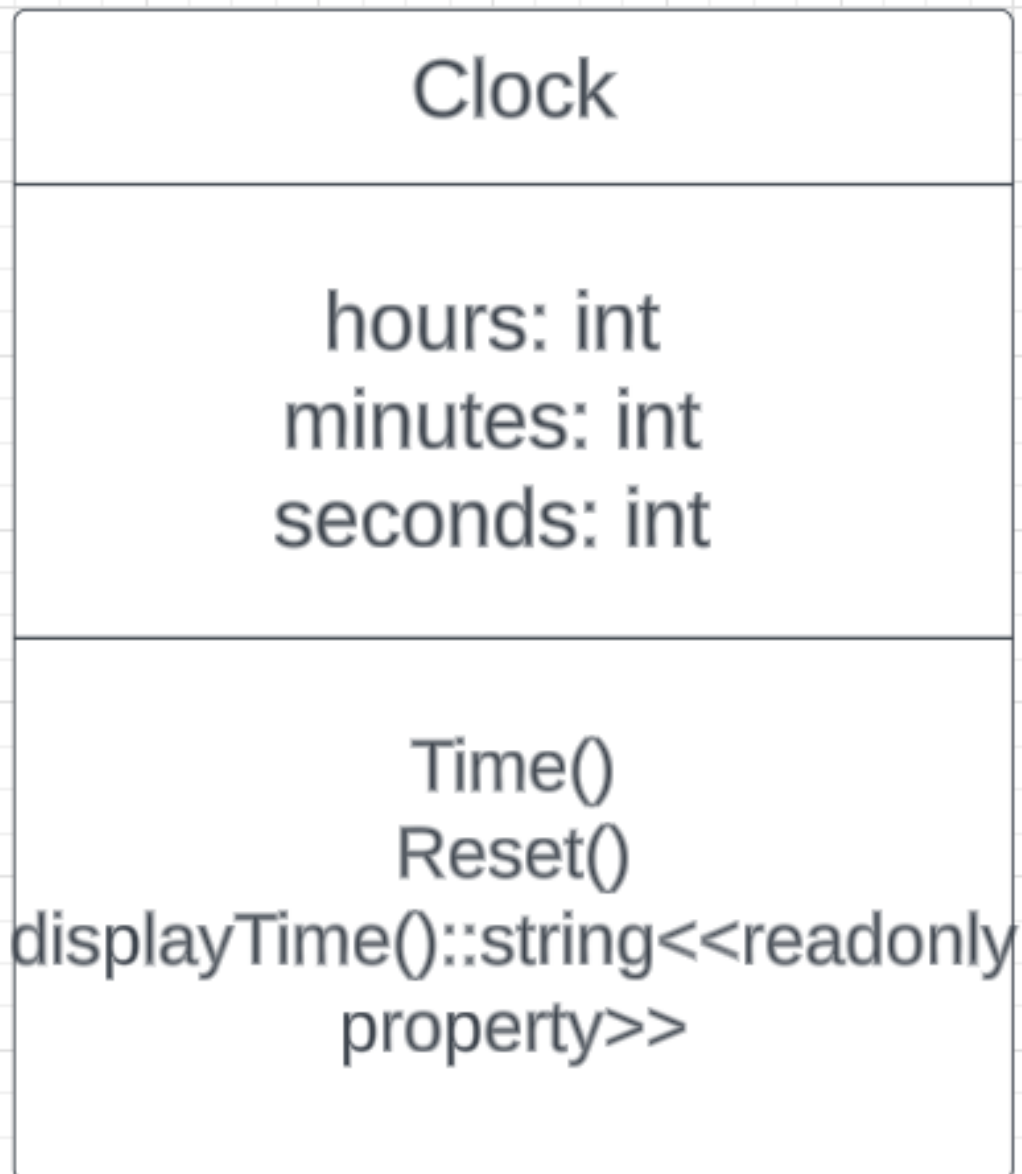


SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Clock Class

PDF generated at 15:57 on Sunday 24th September, 2023



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Clockclass
8  {
9      public class Program
10     {
11         static void Main(string[] args)
12         {
13             Clock clock = new Clock();
14             for (int i = 0; i < 24 * 60 * 60; i++)
15             {
16                 clock.Time();
17                 Console.WriteLine(clock.displayTime());
18             }
19         }
20     }
21 }
```

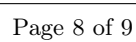
```
1  using System;
2  using System.Collections.Generic;
3  using System.Diagnostics.Metrics;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace Clockclass
9  {
10     public class Clock
11     {
12         private Counterclass hours, minutes, seconds;
13         public Clock()
14         {
15             hours = new Counterclass("Hours");
16             minutes = new Counterclass("Minutes");
17             seconds = new Counterclass("Seconds");
18         }
19         public void Time()
20         {
21             seconds.Increment();
22             if (seconds.Ticks == 60)
23             {
24                 seconds.Reset();
25                 minutes.Increment();
26                 if (minutes.Ticks == 60)
27                 {
28                     minutes.Reset();
29                     hours.Increment();
30                     if (hours.Ticks == 24)
31                     {
32                         Reset();
33                     }
34                 }
35             }
36         }
37         public void Reset()
38         {
39             seconds.Reset();
40             hours.Reset();
41             minutes.Reset();
42         }
43         public string displayTime()
44         {
45             return $"{hours.Ticks:D2}:{minutes.Ticks:D2}:{seconds.Ticks:D2}";
46         }
47     }
48 }
49
50 }
51 }
```

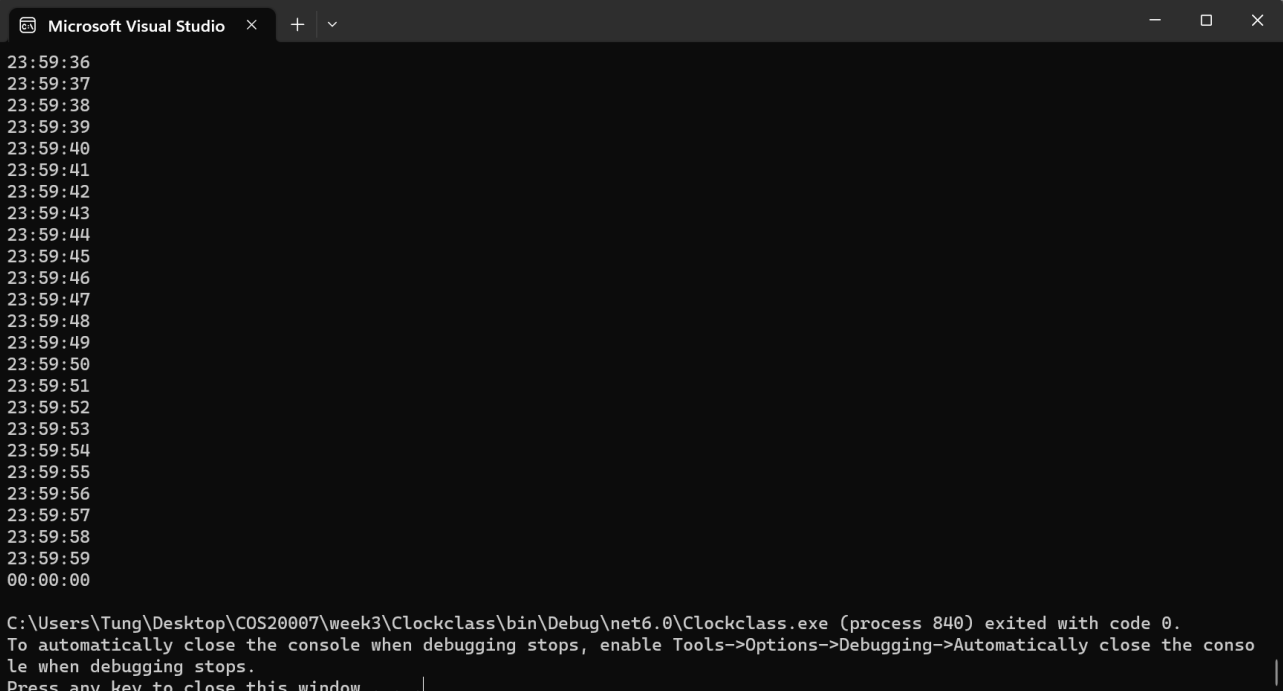
```
1  using NUnit.Framework;
2  using Clockclass;
3
4  namespace ClockTest
5  {
6      [TestFixture]
7      public class Clocktest
8      {
9          private Clock _clockName;
10         [SetUp]
11         public void Setup()
12         {
13             _clockName = new Clock();
14
15         }
16
17         [Test]
18         public void TestFormat()
19         {
20
21             Assert.That(_clockName.displayTime(), Is.EqualTo("00:00:00"));
22         }
23         [Test()]
24         public void TestClockTicks()
25         {
26             for (int i = 0; i < 12; i++)
27             {
28                 _clockName.Time();
29             }
30             Assert.That(_clockName.displayTime(), Is.EqualTo("00:00:12"));
31         }
32         [Test]
33         public void TestSecond()
34         {
35             _clockName.Time();
36             Assert.That(_clockName.displayTime(), Is.EqualTo("00:00:01"));
37         }
38         [Test]
39         public void TestMinute()
40         {
41             for (int i = 0; i < 60; i++)
42             {
43                 _clockName.Time();
44             }
45             Assert.That(_clockName.displayTime(), Is.EqualTo("00:01:00"));
46         }
47         [Test]
48         public void TestHour()
49         {
50             for (int i = 0; i < 3600; i++)
51             {
52                 _clockName.Time();
53             }
54         }
55     }
```

```
54         Assert.That(_clockName.displayTime(), Is.EqualTo("01:00:00"));
55     }
56     [Test]
57     public void TestRest()
58     {
59         for (int i = 0; i == 3600; i++)
60         {
61             _clockName.Time();
62         }
63         _clockName.Reset();
64         Assert.That(_clockName.displayTime(), Is.EqualTo("00:00:00"));
65     }
66 }
67 }
```

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

```
1  using NUnit.Framework;
2  using System.Diagnostics.Metrics;
3  using Clockclass;
4  namespace Countertest;
5
6  [TestFixture]
7  public class Tests
8  {
9      private Counterclass _counter;
10     [SetUp]
11     public void Setup()
12     {
13         _counter = new Counterclass("hour");
14     }
15
16     [Test]
17     public void TestTicks()
18     {
19         Assert.That(_counter.Ticks, Is.EqualTo(0));
20     }
21     [Test()]
22     public void TestIncrement()
23     {
24         _counter.Increment();
25         Assert.That(_counter.Ticks, Is.EqualTo(1));
26     }
27     [Test]
28     public void TestIncrementMutiple()
29     {
30         for (int i = 0; i < 3; i++)
31         {
32             _counter.Increment();
33         }
34         Assert.That(_counter.Ticks, Is.EqualTo(3));
35     }
36     [Test]
37     public void TestReset()
38     {
39         for (int i = 0; i < 8; i++)
40         {
41             _counter.Increment();
42         }
43         _counter.Reset();
44
45         Assert.That(_counter.Ticks, Is.EqualTo(0));
46     }
47 }
48 }
```



```
Microsoft Visual Studio x + v
23:59:36
23:59:37
23:59:38
23:59:39
23:59:40
23:59:41
23:59:42
23:59:43
23:59:44
23:59:45
23:59:46
23:59:47
23:59:48
23:59:49
23:59:50
23:59:51
23:59:52
23:59:53
23:59:54
23:59:55
23:59:56
23:59:57
23:59:58
23:59:59
00:00:00
C:\Users\Tung\Desktop\COS20007\week3\Clockclass\bin\Debug\net6.0\Clockclass.exe (process 840) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|
```