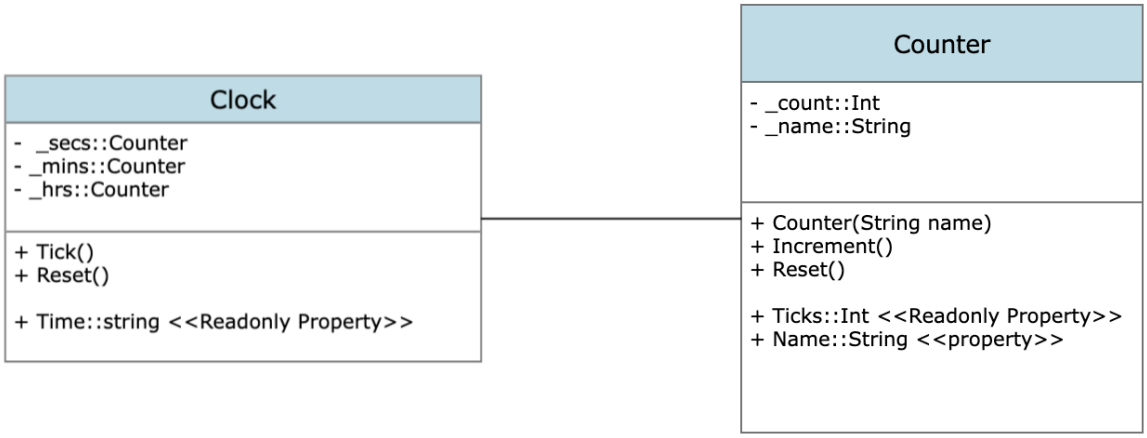
Swinburne University of Technology

COS20007 Object Oriented Programming

3.1P - Clock Class

PDF generated at 21:35 on Sunday 26th March, 2023


Clock
- _secs::Counter
- _mins::Counter
- _hrs::Counter
+ Tick()
+ Reset()
+ Time::string <<Readonly Property>>
Counter
- _count::Int
- _name::String
+ Counter(String name)
+ Increment()
+ Reset()
+ Ticks::Int <<Readonly Property>>
+ Name::String <<property>>

```
namespace ClockClass
{
    class MainClass
    {
        public static void Main(string[] args)
        {
            Clock myClock = new Clock();

            int i = 0;
            while (i < 260)
            {
                myClock.Tick();
                Console.WriteLine(myClock.Time);
                i++;
            }
            Console.ReadLine();
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;


namespace ClockClass
{
    public class Clock
    {
        private Counter _secs;
        private Counter _mins;
        private Counter _hrs;

        public Clock()
        {
            _secs = new Counter("seconds");
            _mins = new Counter("minutes");
            _hrs = new Counter("hours");
        }


        public void Tick()
        {
            _secs.Increment();
            if (_secs.Ticks > 59)
            {
                _mins.Increment();
                _secs.Reset();
            }
            if (_mins.Ticks > 59)
            {
                _hrs.Increment();
                _mins.Reset();
            }
            if (_hrs.Ticks > 23)
            {
                _hrs.Reset();
                _mins.Reset();
                _secs.Reset();
            }
        }

        public void Reset()
        {
            _hrs.Reset();
            _mins.Reset();
            _secs.Reset();
        }

        public string Time
        {
```

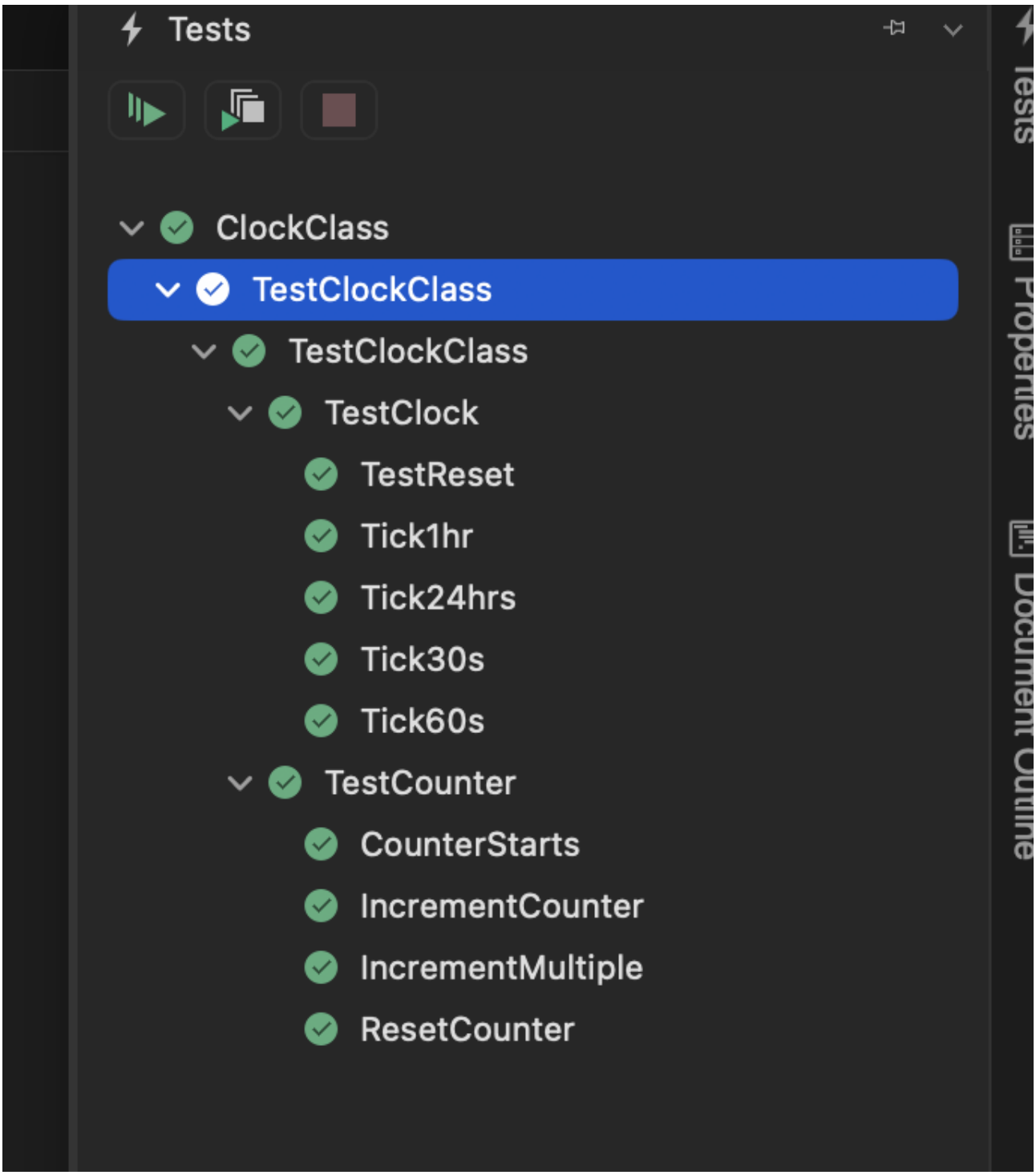
```
54              get
55              {
56                  return
57                      $"{_hrs.Ticks:00}:{_mins.Ticks:00}:{_secs.Ticks:00}";
58              }
59          }
60      }
61  }
```

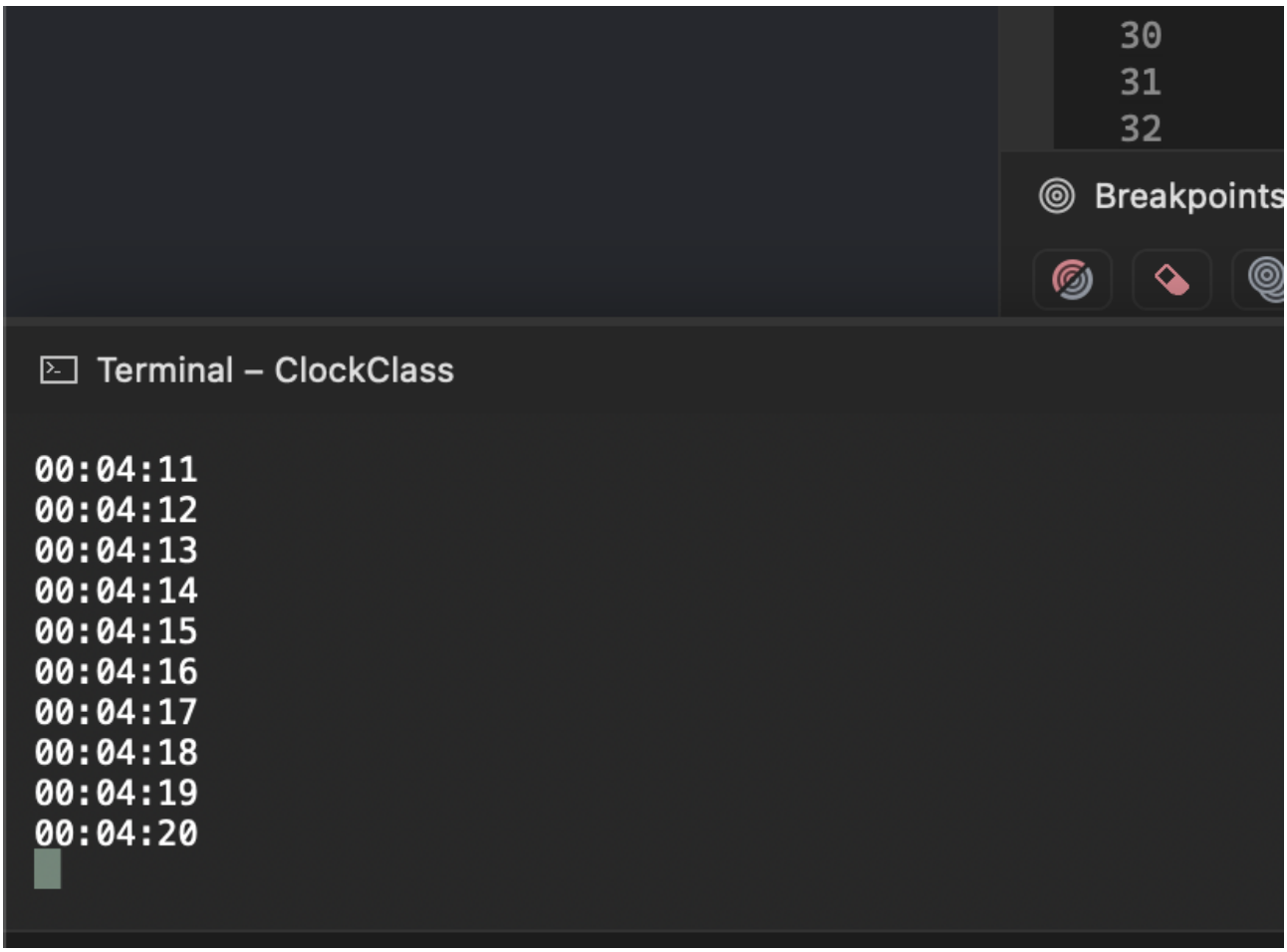
```
using NUnit.Framework;

namespace TestClockClass
{
    public class TestClock
    {
        Clock myClock;
        [SetUp]
        public void Setup()
        {
            myClock = new Clock();
        }


        [Test]
        public void Tick30s()
        {
            for (int i = 0; i < 30; i++)
            {
                myClock.Tick();
            }
            Assert.That(myClock.Time, Is.EqualTo("00:00:30"));
        }

        [Test]
        public void Tick60s()
        {
            for (int i = 0; i < 60; i++)
            {
                myClock.Tick();
            }
            Assert.That(myClock.Time, Is.EqualTo("00:01:00"));
        }

        [Test]
        public void Tick1hr()
        {
            for (int i = 0; i < 3600; i++)
            {
                myClock.Tick();
            }
            Assert.That(myClock.Time, Is.EqualTo("01:00:00"));
        }

        [Test]
        public void Tick24hrs()
        {
            for (int i = 0; i < 86400; i++)
            {
                myClock.Tick();
            }
            Assert.That(myClock.Time, Is.EqualTo("00:00:00"));
        }
```

```
54
55          [Test]
56          public void TestReset()
57          {
58              for (int i = 0; i < 420; i++)
59              {
60                  myClock.Tick();
61              }
62              myClock.Reset();
63              Assert.That(myClock.Time, Is.EqualTo("00:00:00"));
64          }
65
66      }
67  }
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ClockClass
{
    public class Counter
    {
        private int _count;
        private string _name;

        public Counter(string name)
        {
            _name = name;
            _count = 0;
        }

        public int Ticks
        {
            get
            {
                return _count;
            }
        }

        public void Increment()
        {
            _count++;
        }

        public void Reset()
        {
            _count = 0;
        }
    }
}
```

```
using ClockClass;
using System.Diagnostics.Metrics;

namespace TestClockClass
{
    [TestFixture]
    public class TestCounter
    {
        Counter myCounter;

        [SetUp]
        public void Setup()
        {
            myCounter = new Counter("counter");
        }

        [Test]
        public void CounterStarts()
        {
            Assert.That(myCounter.Ticks, Is.EqualTo(0));
        }

        [Test]
        public void IncrementCounter()
        {
            myCounter.Increment();
            Assert.That(myCounter.Ticks, Is.EqualTo(1));
        }

        [Test]
        public void IncrementMultiple()
        {
            for (int i = 0; i < 5; i++)
            {
                myCounter.Increment();
            }
            Assert.That(myCounter.Ticks, Is.EqualTo(5));
        }

        [Test]
        public void ResetCounter()
        {
            myCounter.Increment();
            myCounter.Reset();

            Assert.That(myCounter.Ticks, Is.EqualTo(0));
        }
    }


}
```


Tests
ClockClass
TestClockClass
TestClockClass
TestClock
TestReset
Tick1hr
Tick24hrs
Tick30s
Tick60s
TestCounter
CounterStarts
IncrementCounter
IncrementMultiple
ResetCounter
Tests
Properties
Document Outline


30
31
32
Breakpoints
Terminal – ClockClass
00:04:11
00:04:12
00:04:13
00:04:14
00:04:15
00:04:16
00:04:17
00:04:18
00:04:19
00:04:20