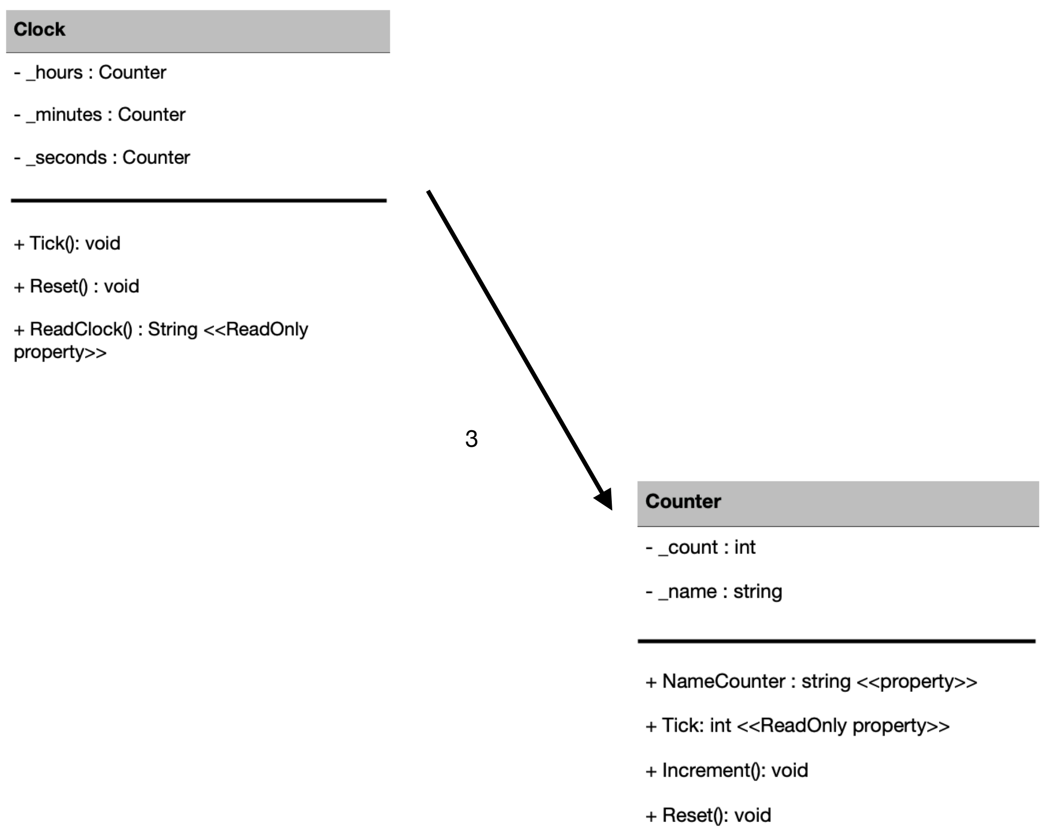
Swinburne University of Technology

COS20007 Object Oriented Programming

Clock Class

PDF generated at 10:26 on Monday 21st August, 2023


Clock
- _hours : Counter
- _minutes : Counter
- _seconds : Counter
+ Tick(): void
+ Reset() : void
+ ReadClock() : String <<ReadOnly property>>
3
Counter
- _count : int
- _name : string
+ NameCounter : string <<property>>
+ Tick: int <<ReadOnly property>>
+ Increment(): void
+ Reset(): void

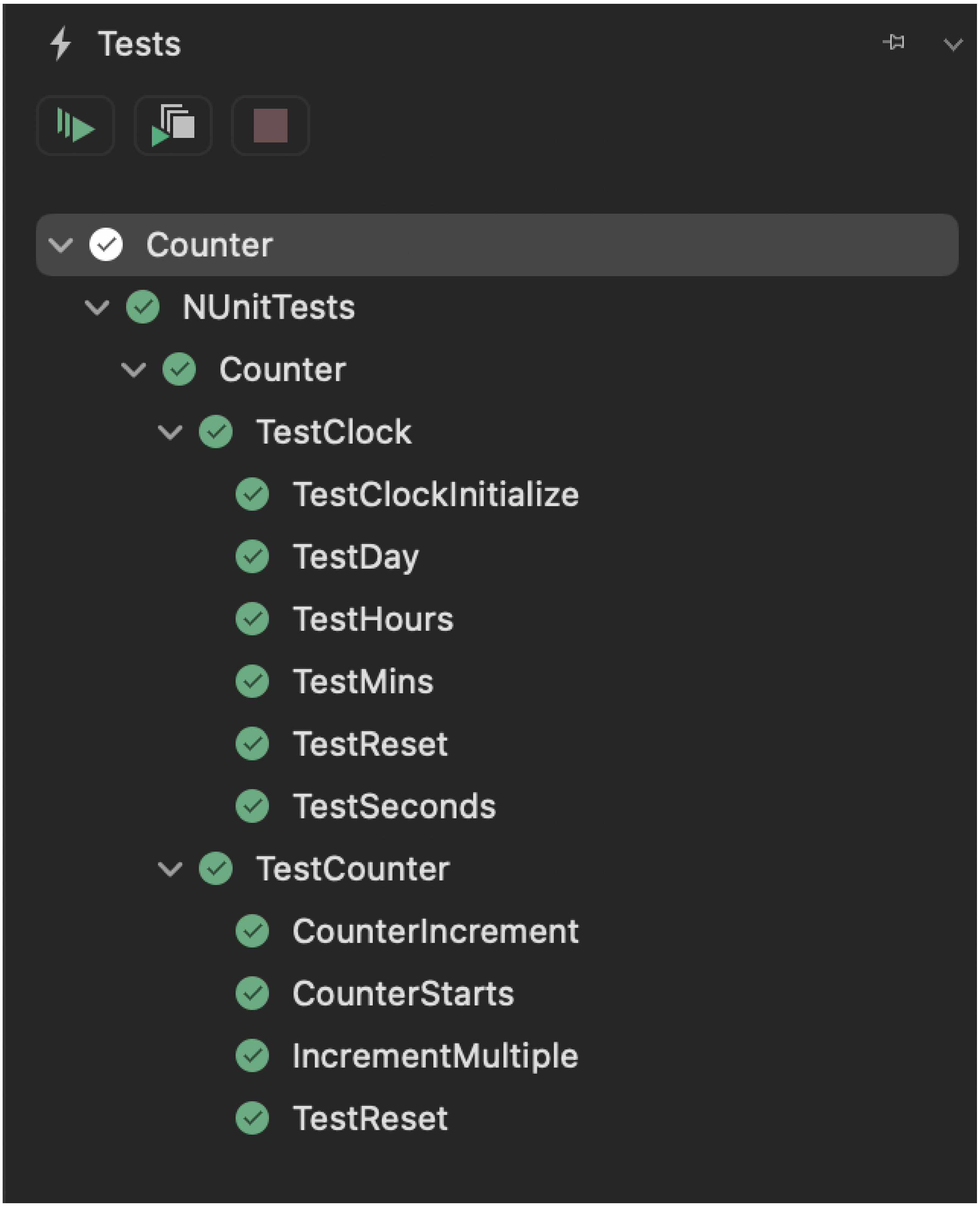
```
using System;

namespace Clock
{
    class Program
    {
        public static void Main(string[] args)
        {
            Clock clock = new Clock();
            while (true)
            {
                clock.Tick();
                Console.WriteLine(clock.ReadClock());
                Thread.Sleep(1000);
            }
        }
    }
}
```

```csharp
using System;

namespace Clock
{
    public class Clock
    {
        // Three counter objects are initialised
        Counter _seconds = new("seconds");
        Counter _minutes = new("minutes");
        Counter _hours = new("hours");

        public void Tick()
        {
            // Increment seconds and handle rollovers for minutes and hours.
            _seconds.Increment();
            if (_seconds.Tick > 59)
            {
                _seconds.Reset();

                _minutes.Increment();
                if (_minutes.Tick > 59)
                {
                    _minutes.Reset();

                    _hours.Increment();
                    if (_hours.Tick > 23)
                    {
                        _hours.Reset();
                    }
                }
            }
        }

        public void Reset()
        {
            // Reset all time components.
            _seconds.Reset();
            _minutes.Reset();
            _hours.Reset();
        }

        public string ReadClock()
        // Formats and returns the time as "hh:mm:ss".
        {
            return _hours.Tick.ToString("00") + ":" + _minutes.Tick.ToString("00") +
 ↪  ":" + _seconds.Tick.ToString("00");
        }
    }
}
```

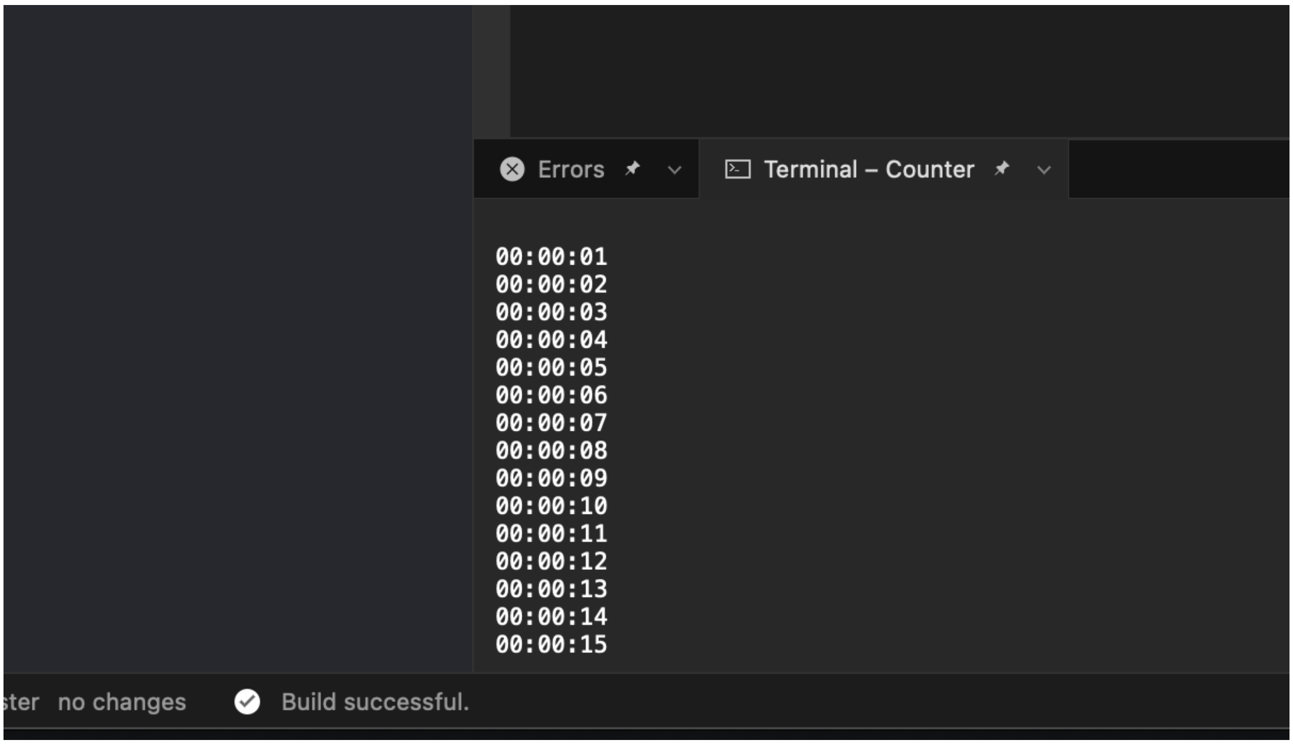
```
using NUnit.Framework;
using Clock;

namespace Counter
{
    [TestFixture]
    public class TestClock
    {
        private Clock.Clock _testClock;

        [SetUp]
        public void Setup()
        {
            _testClock = new Clock.Clock();
        }

        [Test]
        public void TestClockInitialize()
        {
            Assert.That(_testClock.ReadClock(), Is.EqualTo("00:00:00"));
        }

        [Test]
        public void TestSeconds()
        {
            _testClock.Tick();
            Assert.That(_testClock.ReadClock(), Is.EqualTo("00:00:01"));
        }

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

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

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

        [Test]
        public void TestReset()
        {
            _testClock.Tick();
            _testClock.Reset();

            Assert.That(_testClock.ReadClock(), Is.EqualTo("00:00:00"));
        }
    }
}
```

```
using System;

namespace Clock
{
    public class Counter
    {
        //the fields enable the counter to know its count and name values
        private int _count;
        private string _name;

        public Counter()
        {
        }

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

        public string NameCounter
        {
            //get method is read only
            get
            {
                return _name;
            }
            //set method is write only
            set
            {
                _name = value;
            }
        }

        public int Tick
        {
            get
            {
                return _count;
            }
        }

        public void Increment()
        {
            _count += 1;
        }

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

```
using NUnit.Framework;
using Clock;

namespace Counter
{
    [TestFixture]
    public class TestCounter
    {
        private Clock.Counter _counter;

        [SetUp]
        public void SetUp()
        {
            _counter = new Clock.Counter();
        }

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

        [Test]
        public void CounterIncrement()
        {
            _counter.Increment();
            Assert.That(_counter.Tick, Is.EqualTo(1));
        }

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

        [Test]
        public void TestReset()
        {
            _counter.Increment();
            _counter.Reset();
            Assert.That(_counter.Tick, Is.EqualTo(0));
        }
    }
}
```



```
✓ Counter
  ✓ NUnitTests
    ✓ Counter
      ✓ TestClock
        ✓ TestClockInitialize
        ✓ TestDay
        ✓ TestHours
        ✓ TestMins
        ✓ TestReset
        ✓ TestSeconds
      ✓ TestCounter
        ✓ CounterIncrement
        ✓ CounterStarts
        ✓ IncrementMultiple
        ✓ TestReset
```



```
00:00:01
00:00:02
00:00:03
00:00:04
00:00:05
00:00:06
00:00:07
00:00:08
00:00:09
00:00:10
00:00:11
00:00:12
00:00:13
00:00:14
00:00:15
```

ster no changes Build successful.