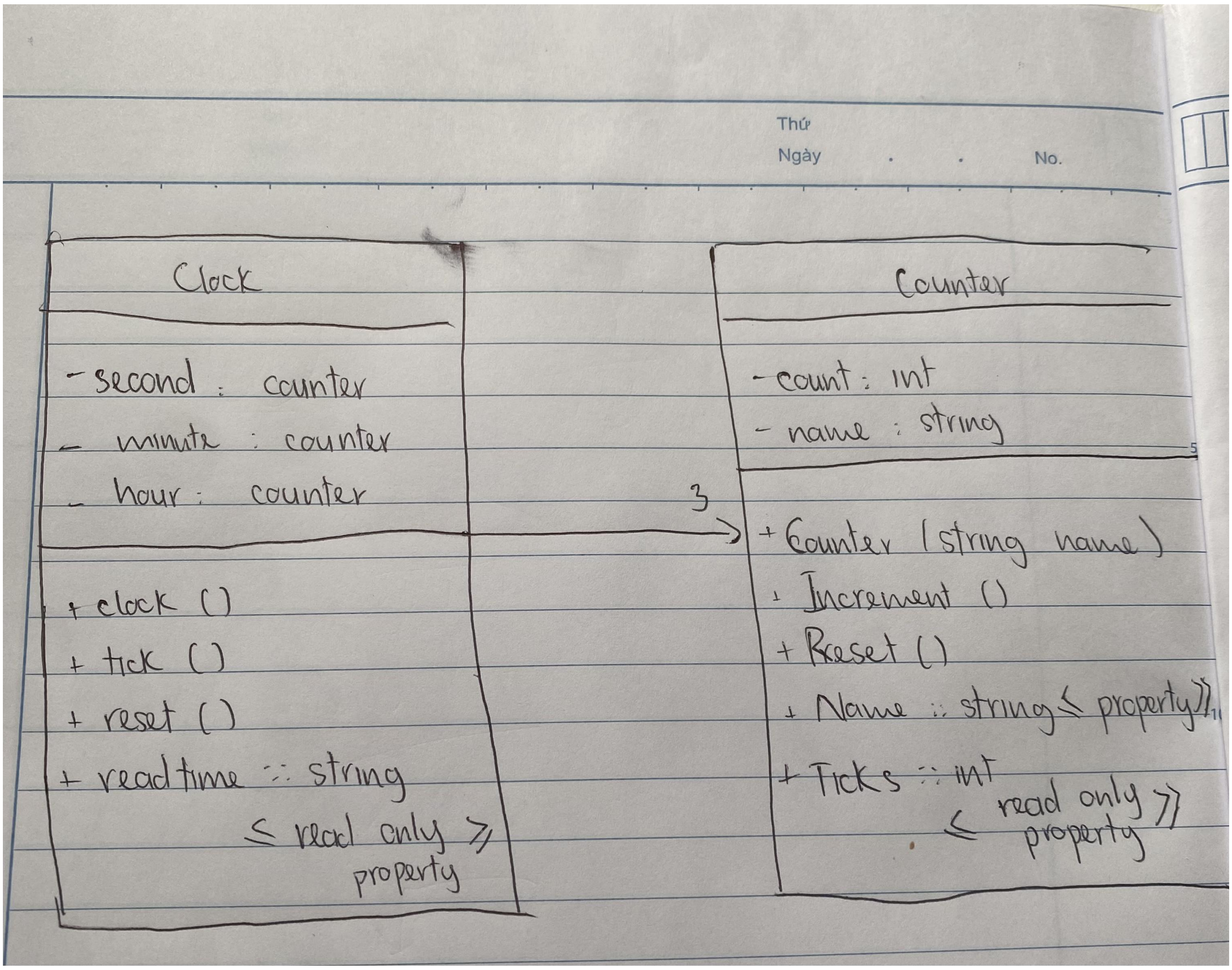
Swinburne University of Technology

COS20007 Object Oriented Programming

Clock Class

PDF generated at 14:10 on Thursday 31st August, 2023

Produced by OOP Submission Formatter


Thứ
Ngày
No.
Clock
- second : counter
- minute : counter
- hour : counter
+ clock ()
+ tick ()
+ reset ()
+ readtime :: string
≤ read only ≥ property
3
Counter
- count : int
- name : string
+ Counter (string name)
+ Increment ()
+ Reset ()
+ Name :: string ≤ property ≥
+ Ticks :: int
≤ read only property ≥

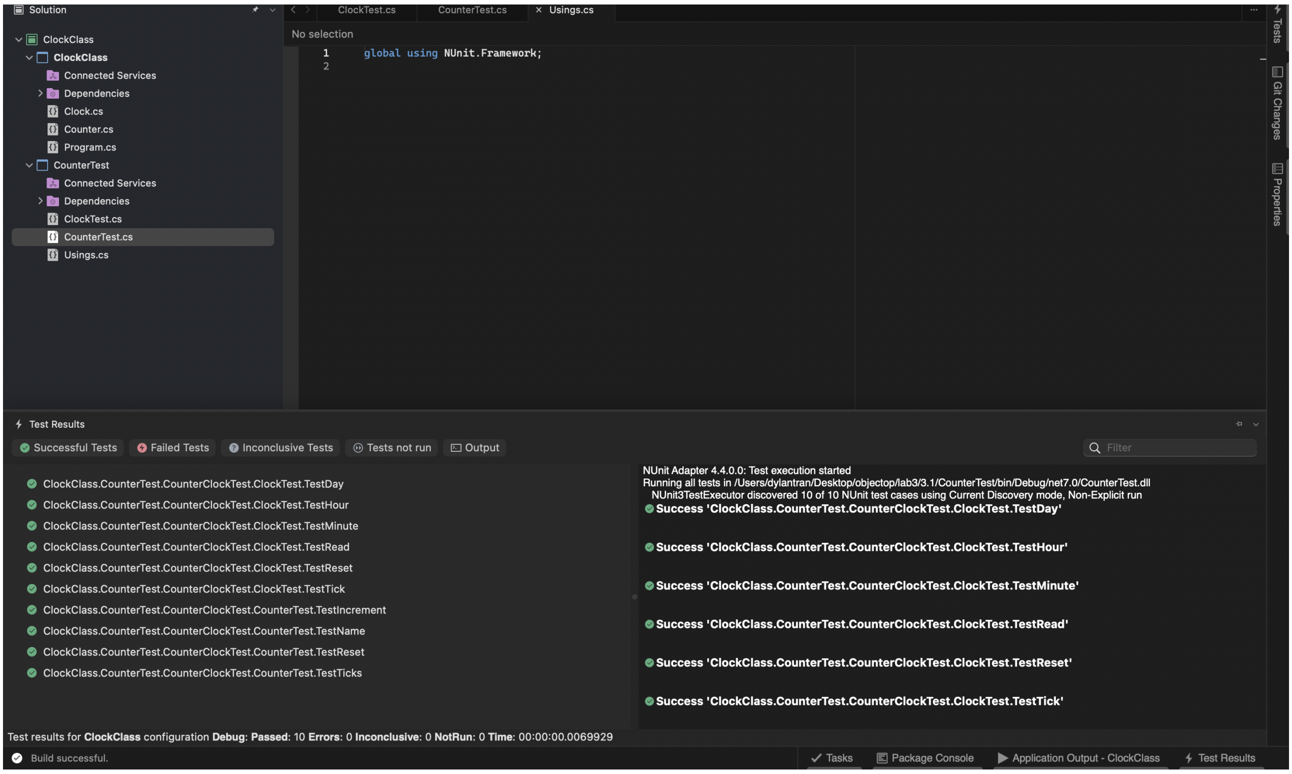
```
using System;

namespace CounterClock
{
    class Program
    {
        public static void Main(string[] args)
        {
            Clock clock = new Clock();
            for (int i = 0; i < 60 * 60 * 24; i++)
            {
                Console.WriteLine(clock.ReadTime);
                clock.Tick();
            }
        }
    }
}
```

```
using System;

namespace CounterClock
{
    public class Clock
    {
        private Counter _seconds;
        private Counter _minutes;
        private Counter _hours;

        public Clock()
        {
            _seconds = new Counter("seconds");
            _minutes = new Counter("minutes");
            _hours = new Counter("hours");
        }

        public void Reset()
        {
            _seconds.Reset();
            _minutes.Reset();
            _hours.Reset();
        }

        public string ReadTime
        {
            get
            {
                return $"{_hours.Ticks:D2}:{_minutes.Ticks:D2}:{_seconds.Ticks:D2}";
            }
        }

        public void Tick()
        {
            if (_seconds.Ticks < 59)
            {
                _seconds.Increment();
            }
            else
            {
                _seconds.Reset();
                if (_minutes.Ticks < 59)
                {
                    _minutes.Increment();
                }
                else
                {
                    _minutes.Reset();
                    if (_hours.Ticks == 23)
                    {
                        _hours.Reset();
                    }
                    else
                    {
                        _hours.Increment();
                    }
                }
            }
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using NUnit.Framework;
using CounterClock;

namespace CounterClockTest
{
    public class ClockTest
    {
        private Clock _clock;

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

        [Test]
        public void TestReset()
        {
            _clock.Tick();
            _clock.Reset();
            Assert.That(_clock.ReadTime, Is.EqualTo("00:00:00"), "Test Reset");
        }

        [Test]
        public void TestRead()
        {
            for (int i = 0; i < 5402; i++)
            {
                _clock.Tick();
            }
            Assert.That(_clock.ReadTime, Is.EqualTo("01:30:02"), "Test Read");
        }

        [Test]
        public void TestTick()
        {
            _clock.Tick();
            Assert.That(_clock.ReadTime, Is.EqualTo("00:00:01"), "Test Tick");
        }

        [Test]
        public void TestMinute()
        {
            for (int i = 0; i < 60; i++)
            {
                _clock.Tick();
            }
            Assert.That(_clock.ReadTime, Is.EqualTo("00:01:00"), "Test Minute");
        }
        [Test]
        public void TestHour()
        {
            for (int i = 0; i < 60*60; i++)
            {
                _clock.Tick();
            }
            Assert.That(_clock.ReadTime, Is.EqualTo("01:00:00"), "Test Hour");
        }

        [Test]
        public void TestDay()
        {
            for (int i = 0; i < 60 * 60 * 24; i++)
            {
                _clock.Tick();
            }
            Assert.That(_clock.ReadTime, Is.EqualTo("00:00:00"), "Test Day");
        }
    }
}
```

```
using System;

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

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

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

        public void Reset()
        {
            _count = 0;
        }

        public string Name
        {
            get
            {
                return _name;
            }
            set
            {
                _name = value;
            }
        }

        public int Ticks
        {
            get
            {
                return _count;
            }
        }
    }
}
```

```
using System;
using NUnit.Framework;
using CounterClock;


namespace CounterClockTest
{
    public class CounterTest
    {
        private Counter _counter;

        [SetUp]
        public void Setup()
        {
            _counter = new Counter("TestCounter");
        }

        [Test]
        public void TestIncrement()
        {
            _counter.Increment();
            Assert.That(_counter.Ticks, Is.EqualTo(1), "Test Increment");
        }

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

        [Test]
        public void TestName()
        {
            Assert.That(_counter.Name, Is.EqualTo("TestCounter"), "Test Name Get");
            _counter.Name = "Test";
            Assert.That(_counter.Name, Is.EqualTo("Test"), "Test Name Set");
        }

        [Test]
        public void TestTicks()
        {
            Assert.That(_counter.Ticks, Is.EqualTo(0), "Test Ticks");

            _counter.Increment();
            Assert.That(_counter.Ticks, Is.EqualTo(1), "Test Ticks after using
↪   Increment");

            _counter.Reset();
            Assert.That(_counter.Ticks, Is.EqualTo(0), "Test Ticks after using
↪   Increment Reser");
        }
    }
}
```

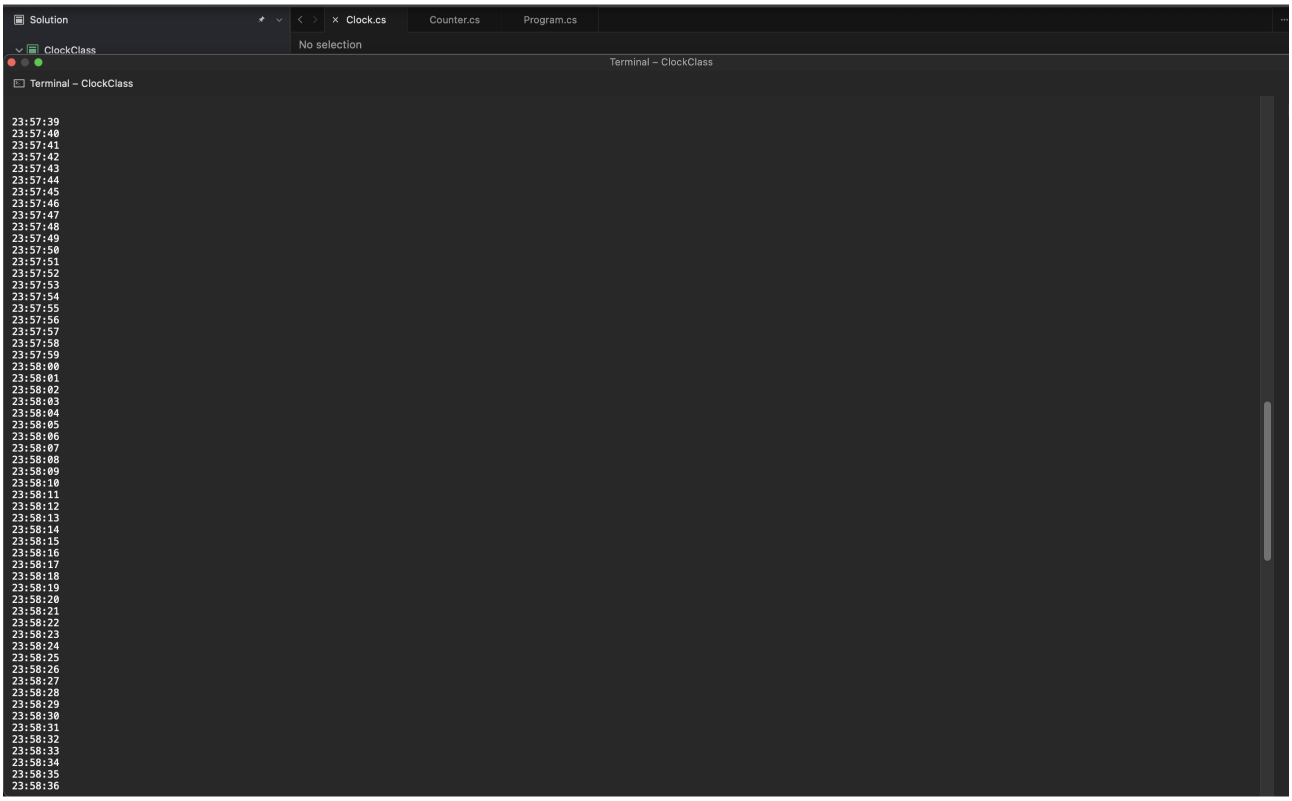


```
global using NUnit.Framework;
```

```
ClockClass.CounterTest.CounterClockTest.ClockTest.TestDay
ClockClass.CounterTest.CounterClockTest.ClockTest.TestHour
ClockClass.CounterTest.CounterClockTest.ClockTest.TestMinute
ClockClass.CounterTest.CounterClockTest.ClockTest.TestRead
ClockClass.CounterTest.CounterClockTest.ClockTest.TestReset
ClockClass.CounterTest.CounterClockTest.ClockTest.TestTick
ClockClass.CounterTest.CounterClockTest.CounterTest.TestIncrement
ClockClass.CounterTest.CounterClockTest.CounterTest.TestName
ClockClass.CounterTest.CounterClockTest.CounterTest.TestReset
ClockClass.CounterTest.CounterClockTest.CounterTest.TestTicks
```

```
NUnit Adapter 4.4.0.0: Test execution started
Running all tests in /Users/dylantran/Desktop/objectop/lab3/3.1/CounterTest/bin/Debug/net7.0/CounterTest.dll
  NUnit3TestExecutor discovered 10 of 10 NUnit test cases using Current Discovery mode, Non-Explicit run
Success 'ClockClass.CounterTest.CounterClockTest.ClockTest.TestDay'

Success 'ClockClass.CounterTest.CounterClockTest.ClockTest.TestHour'

Success 'ClockClass.CounterTest.CounterClockTest.ClockTest.TestMinute'

Success 'ClockClass.CounterTest.CounterClockTest.ClockTest.TestRead'

Success 'ClockClass.CounterTest.CounterClockTest.ClockTest.TestReset'

Success 'ClockClass.CounterTest.CounterClockTest.ClockTest.TestTick'
```

```
Test results for ClockClass configuration Debug: Passed: 10 Errors: 0 Inconclusive: 0 NotRun: 0 Time: 00:00:00.0069929
Build successful.
```



```
23:57:39
23:57:40
23:57:41
23:57:42
23:57:43
23:57:44
23:57:45
23:57:46
23:57:47
23:57:48
23:57:49
23:57:50
23:57:51
23:57:52
23:57:53
23:57:54
23:57:55
23:57:56
23:57:57
23:57:58
23:57:59
23:58:00
23:58:01
23:58:02
23:58:03
23:58:04
23:58:05
23:58:06
23:58:07
23:58:08
23:58:09
23:58:10
23:58:11
23:58:12
23:58:13
23:58:14
23:58:15
23:58:16
23:58:17
23:58:18
23:58:19
23:58:20
23:58:21
23:58:22
23:58:23
23:58:24
23:58:25
23:58:26
23:58:27
23:58:28
23:58:29
23:58:30
23:58:31
23:58:32
23:58:33
23:58:34
23:58:35
23:58:36
```