Swinburne University of Technology

Object Oriented Programming (2022 S1)

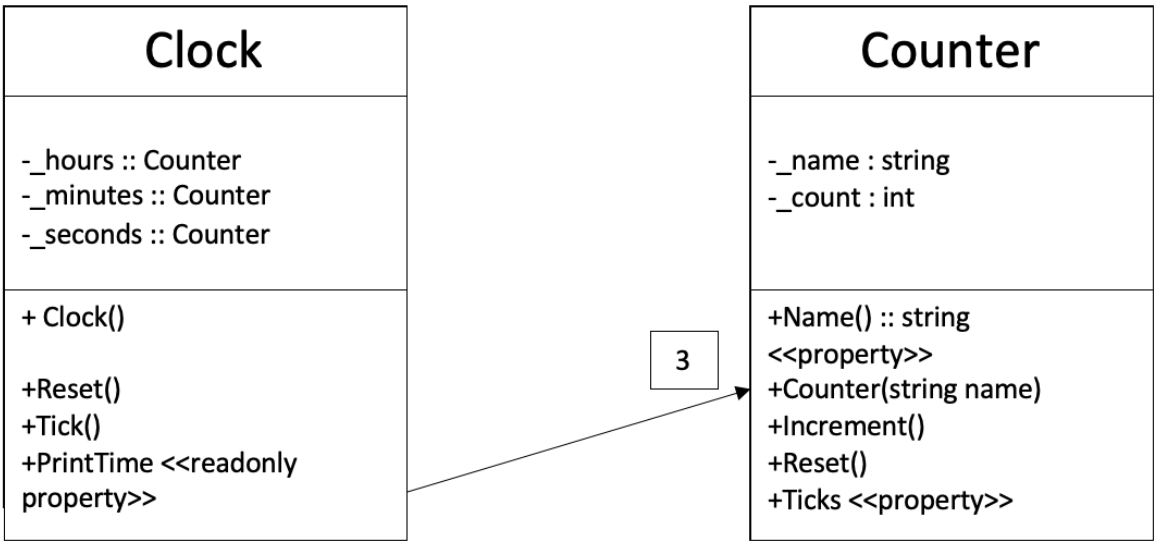
Doubtfire Submission

Task 3.3P: Clock Class

Submitted By:
Vaissheenavi Prabakaran
103508183
2022/04/22 20:02

Tutor:
Jai Cornes

April 22, 2022


Clock
-_hours :: Counter
-_minutes :: Counter
-_seconds :: Counter
+ Clock()
+Reset()
+Tick()
+PrintTime <<readonly property>>
3
Counter
-_name : string
-_count : int
+Name() :: string <<property>>
+Counter(string name)
+Increment()
+Reset()
+Ticks <<property>>

```
using System;
namespace ClockTask
{
    public class Program
    {
        public static void Main(string[] args)
        {
            Clock clock = new Clock();

            //string readtime = "";

            for (int i = 0; i < 87000; i++)
            {
                clock.Tick();
                Console.WriteLine("Clock time is: " + clock.PrintTime);
            }

            //Console.WriteLine("Clock time is: " + clock.PrintTime);
        }
    }
}
```

```
using System;
namespace ClockTask
{
    public class Clock
    {
        private Counter _hours;
        private Counter _minutes;
        private Counter _seconds;


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

        public void Tick()
        {
            _seconds.Increment();
            if (_seconds.Ticks == 60)
            {
                _seconds.Reset();
                _minutes.Increment();
            }

            if (_minutes.Ticks == 60)
            {
                _minutes.Reset();
                _hours.Increment();
            }

             if (_hours.Ticks == 24)
             {
                _hours.Reset();
             }

        }



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


        public string PrintTime
        {
            get
            {
```

```
54                  return _hours.Ticks.ToString("00") + ":" +
                     ↪  _minutes.Ticks.ToString("00") + ":" +
                     ↪  _seconds.Ticks.ToString("00");
55
56              }
57
58          }
59
60
61
62
63      }
64  }
```

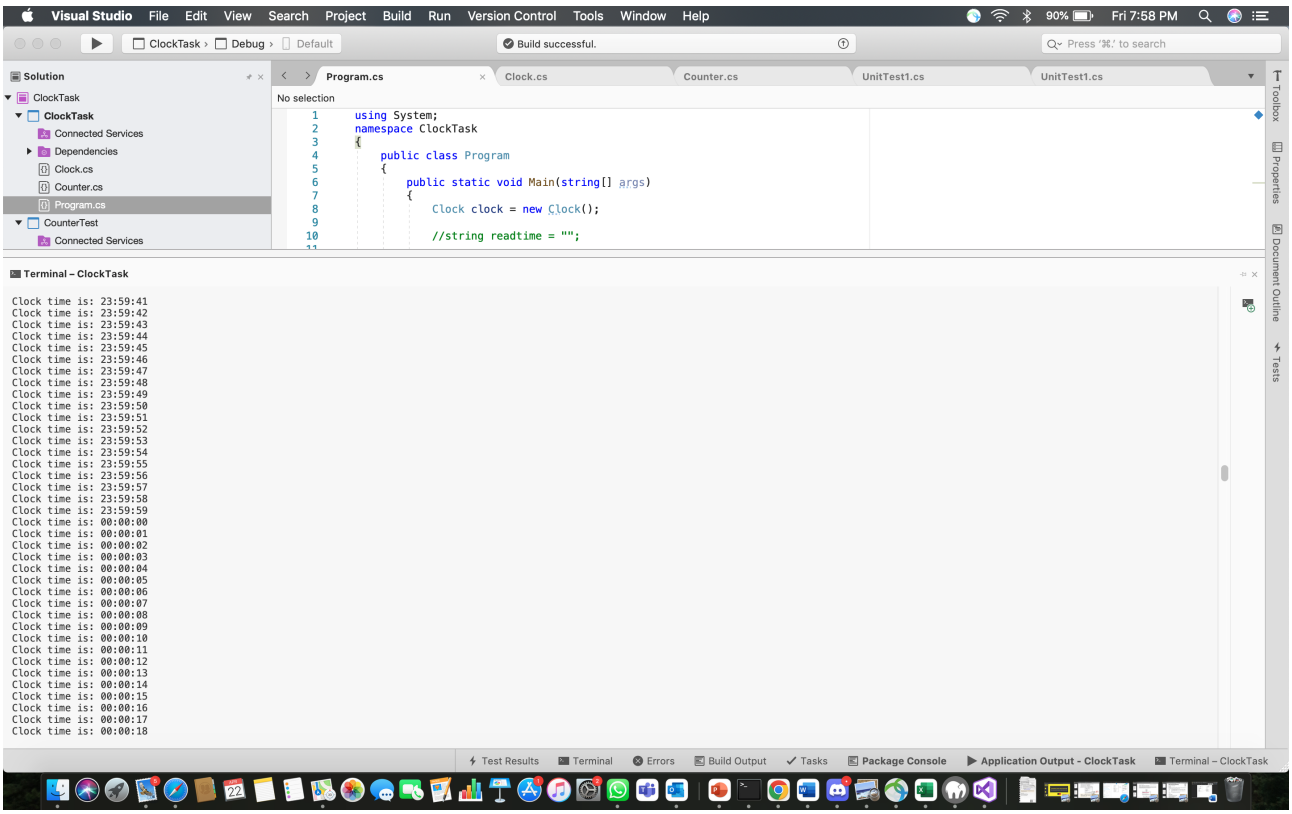
```
using System;
namespace ClockTask
{
    public class Counter
    {
        private int _count;
        private string _name;

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

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

        public void Increment()
        {
            _count++;
        }

        public void Reset()
        {
            _count = 0;

        }


        public int Ticks
        {
            get
            {
                return _count;
            }

            set
            {
                _count = value;
            }

        }
    }
}
```

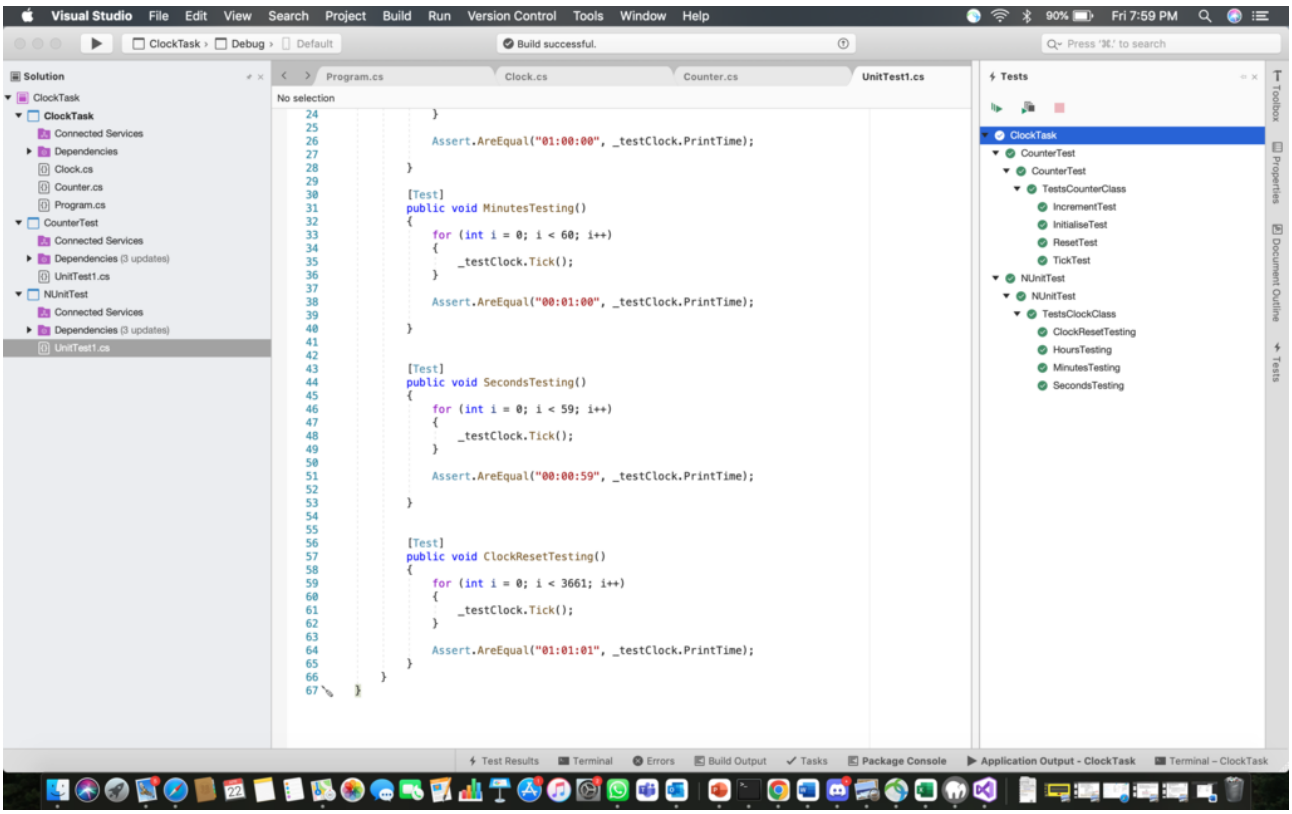
```
using NUnit.Framework;
using ClockTask;

namespace NUnitTest
{
    public class TestsClockClass
    {
        private Clock _testClock;


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


        [Test]
        public void HoursTesting()
        {
            for (int i = 0; i < 3600; i++)
            {
                _testClock.Tick();
            }

            Assert.AreEqual("01:00:00", _testClock.PrintTime);

        }

        [Test]
        public void MinutesTesting()
        {
            for (int i = 0; i < 60; i++)
            {
                _testClock.Tick();
            }

            Assert.AreEqual("00:01:00", _testClock.PrintTime);

        }


        [Test]
        public void SecondsTesting()
        {
            for (int i = 0; i < 59; i++)
            {
                _testClock.Tick();
            }

            Assert.AreEqual("00:00:59", _testClock.PrintTime);

        }
```

```
54
55
56          [Test]
57          public void ClockResetTesting()
58          {
59              for (int i = 0; i < 3661; i++)
60              {
61                  _testClock.Tick();
62              }
63
64              Assert.AreEqual("01:01:01", _testClock.PrintTime);
65          }
66      }
67  }
```

```
using NUnit.Framework;
using ClockTask;

namespace CounterTest
{
    public class TestsCounterClass
    {
        private Counter _testCounter;


        [SetUp]
        public void Setup()
        {
            _testCounter = new Counter("testCounter");
        }

        [Test]
        public void ResetTest()
        {
            ClockTask.Counter testCounter = new ClockTask.Counter("First Counter");

            testCounter.Reset();

            Assert.AreEqual(0, testCounter.Ticks);
        }


        [Test]
        public void InitialiseTest()
        {
            ClockTask.Counter testCounter = new ClockTask.Counter("First Counter");

            int expect = 0;
            int actual = testCounter.Ticks;

            Assert.AreEqual(expect, actual);
        }


        [Test]
        public void IncrementTest()
        {
            ClockTask.Counter testCounter = new ClockTask.Counter("First Counter");

            testCounter.Increment();

            testCounter.Increment();

            Assert.AreEqual(2, testCounter.Ticks);
        }


        [Test]
```

```
54          public void TickTest()
55          {
56              ClockTask.Counter testCounter = new ClockTask.Counter("First Counter");
57
58              testCounter.Ticks = 5;
59
60              Assert.AreNotEqual(4, testCounter.Ticks);
61          }
62      }
63  }
```


Visual Studio File Edit View Search Project Build Run Version Control Tools Window Help
90% Fri 7:58 PM
ClockTask › Debug › Default
Build successful.
Press '⌘,' to search
Solution
ClockTask
ClockTask
Connected Services
Dependencies
Clock.cs
Counter.cs
Program.cs
CounterTest
Connected Services
Program.cs
Clock.cs
Counter.cs
UnitTest1.cs
UnitTest1.cs
No selection
1 using System;
2 namespace ClockTask
3 {
4 public class Program
5 {
6 public static void Main(string[] args)
7 {
8 Clock clock = new Clock();
9
10 //string readtime = "";
Toolbox
Properties
Document Outline
Tests
Terminal – ClockTask
Clock time is: 23:59:41
Clock time is: 23:59:42
Clock time is: 23:59:43
Clock time is: 23:59:44
Clock time is: 23:59:45
Clock time is: 23:59:46
Clock time is: 23:59:47
Clock time is: 23:59:48
Clock time is: 23:59:49
Clock time is: 23:59:50
Clock time is: 23:59:51
Clock time is: 23:59:52
Clock time is: 23:59:53
Clock time is: 23:59:54
Clock time is: 23:59:55
Clock time is: 23:59:56
Clock time is: 23:59:57
Clock time is: 23:59:58
Clock time is: 23:59:59
Clock time is: 00:00:00
Clock time is: 00:00:01
Clock time is: 00:00:02
Clock time is: 00:00:03
Clock time is: 00:00:04
Clock time is: 00:00:05
Clock time is: 00:00:06
Clock time is: 00:00:07
Clock time is: 00:00:08
Clock time is: 00:00:09
Clock time is: 00:00:10
Clock time is: 00:00:11
Clock time is: 00:00:12
Clock time is: 00:00:13
Clock time is: 00:00:14
Clock time is: 00:00:15
Clock time is: 00:00:16
Clock time is: 00:00:17
Clock time is: 00:00:18
Test Results Terminal Errors Build Output Tasks Package Console Application Output - ClockTask Terminal – ClockTask


Visual Studio File Edit View Search Project Build Run Version Control Tools Window Help
Build successful.
Solution
ClockTask
Connected Services
Dependencies
Clock.cs
Counter.cs
Program.cs
CounterTest
Connected Services
Dependencies (3 updates)
UnitTest1.cs
NUnitTest
Connected Services
Dependencies (3 updates)
UnitTest1.cs
Program.cs
Clock.cs
Counter.cs
UnitTest1.cs
No selection
Assert.AreEqual("01:00:00", _testClock.PrintTime);
[Test]
public void MinutesTesting()
for (int i = 0; i < 60; i++)
_testClock.Tick();
Assert.AreEqual("00:01:00", _testClock.PrintTime);
[Test]
public void SecondsTesting()
for (int i = 0; i < 59; i++)
_testClock.Tick();
Assert.AreEqual("00:00:59", _testClock.PrintTime);
[Test]
public void ClockResetTesting()
for (int i = 0; i < 3661; i++)
_testClock.Tick();
Assert.AreEqual("01:01:01", _testClock.PrintTime);
Tests
ClockTask
CounterTest
CounterTest
TestsCounterClass
IncrementTest
InitialiseTest
ResetTest
TickTest
NUnitTest
NUnitTest
TestsClockClass
ClockResetTesting
HoursTesting
MinutesTesting
SecondsTesting
Test Results
Terminal
Errors
Build Output
Tasks
Package Console
Application Output - ClockTask
Terminal – ClockTask