COS20007 - Object Oriented Programming

Student name: Nguyen Duc Manh
ID: 105547489

3.1P - Clock Class with your own hour format

```
using System;
using SplashKitSDK;

namespace CounterTask
{
    class Program
    {
        static void Main(string[] args)
        {
            Clock myClock = new Clock();

            for (int i = 0; i < 86400; i ++)
            {
                //Thread.Sleep(10); //latency
                //Console.Clear();
                myClock.TimeIncrease();
                Console.WriteLine(myClock.ClockDisplay());
            }
            Console.WriteLine("This is the end of my Clock program...");
        }
    }
}
```

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

namespace CounterTask
{
    public class Counter
    {
        // fields
        int _count;
        string _name;

        public Counter(string name, int count) //Constructor
        {
            _name = name;
            _count = count;
        }

        //methods
        public void Increment()
        {
            _count++;
        }

        public void Reset()
        {
            _count = 0;
        }
        public void SetCount(int value)
        {
            _count = value;
        }

        //public void ResetByDefault()
        //{
        //    unchecked
        //    {
        //        _count = (int)2147483647489; //105547489
        //    }
        //}

        //properties
        public string Name
        {
            get
            {
```

```
                return _name;
            }
            set
            {
                _name = value;
            }
        }

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

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

namespace CounterTask
{
    public class Clock
    {
        //field
        Counter _second = new Counter("second", 0);
        Counter _minute = new Counter("minute", 0);
        Counter _hour = new Counter("hour", 12);
        string _period = "AM";

        //constructor

        //method
        public void TimeIncrease()
        {
            _second.Increment();

            if (_second.Ticks > 59)
            {
                _second.Reset();
                _minute.Increment();

                if (_minute.Ticks > 59)
                {
                    _minute.Reset();
                    _hour.Increment();

                    if (_hour.Ticks > 12) // 12-hour format
                    {
                        _hour.SetCount(1);
                    }
                    else if (_hour.Ticks == 12 && _minute.Ticks == 0 &&
                      _second.Ticks == 0)
                    {
                        _period = _period == "AM" ? "PM" : "AM";
                    }
                }
            }
        }

        public string ClockDisplay()
        {
            return $"{_hour.Ticks:D2}:{_minute.Ticks:D2}:{_second.Ticks:D2}
```

```
                {_period}";
        }
    }
}
```


Microsoft Visual Studio Debug Console
11:59:37 PM
11:59:38 PM
11:59:39 PM
11:59:40 PM
11:59:41 PM
11:59:42 PM
11:59:43 PM
11:59:44 PM
11:59:45 PM
11:59:46 PM
11:59:47 PM
11:59:48 PM
11:59:49 PM
11:59:50 PM
11:59:51 PM
11:59:52 PM
11:59:53 PM
11:59:54 PM
11:59:55 PM
11:59:56 PM
11:59:57 PM
11:59:58 PM
11:59:59 PM
12:00:00 AM
This is the end of my Clock program...
C:\msys64\home\Bill\CounterTask 2\bin\Debug\net8.0\CounterTask.exe (process 13176) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .

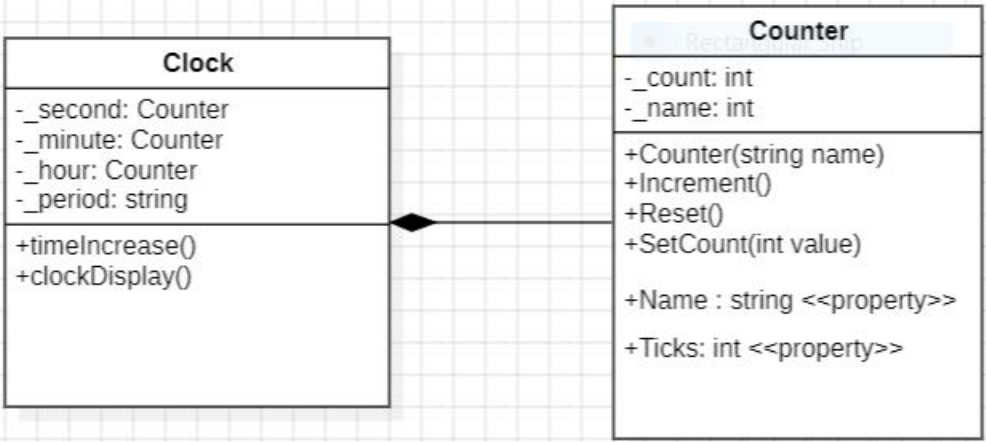
```
namespace CounterTask
{
    public class CounterT
    {
        Counter _testCounter;
        private int i;

        [SetUp]
        public void Setup()
        {
            _testCounter = new Counter("test", 0);
        }

        [Test]
        public void CounterInitial0()
        {
            Assert.That(_testCounter.Ticks, Is.EqualTo(0));
        }

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

        [Test]
        public void MultipleIncrementTest()
        {
            for (i = 0; i < 10; i ++)
            {
                _testCounter.Increment();
            }
            Assert.That(_testCounter.Ticks, Is.EqualTo(10));
        }

        [Test]
        public void ResettingTest()
        {
            for (i = 0; i < 10; i++)
            {
                _testCounter.Increment();
            }
            _testCounter.Reset();
            Assert.That(_testCounter.Ticks, Is.EqualTo(0));
        }
    }
}
```

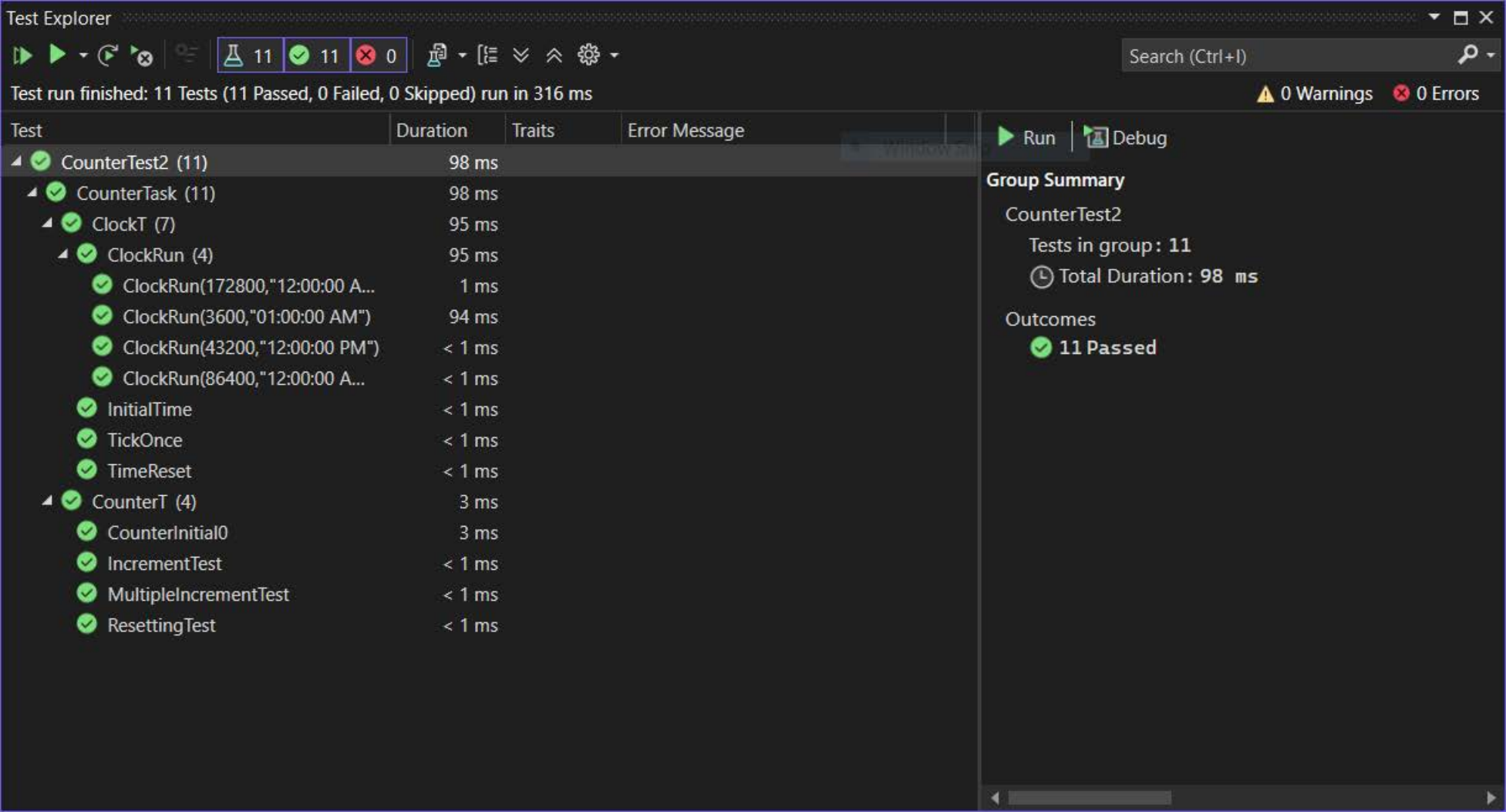
```
namespace CounterTask
{
    public class ClockT
    {
        Clock _testClock;
        string _startTime;

        [SetUp]
        public void Setup()
        {
            _testClock = new Clock();
            _startTime = "12:00:00 AM";
        }

        [Test]
        public void InitialTime()
        {
            Assert.That(_startTime, Is.EqualTo(_testClock.ClockDisplay()));
        }

        [Test]
        public void TickOnce()
        {
            _testClock.TimeIncrease();
            Assert.That(_testClock.ClockDisplay(), Is.EqualTo("12:00:01 AM"));
        }

        [Test]
        public void TimeReset()
        {
            for (int i = 0; i < 86400; i ++)
            {
                _testClock.TimeIncrease();
            }
            Assert.That(_testClock.ClockDisplay(), Is.EqualTo("12:00:00 AM"));
        }

        [TestCase (3600, "01:00:00 AM")] //1 hour
        [TestCase (43200, "12:00:00 PM")] //12 hours
        [TestCase(86400, "12:00:00 AM")] //24 hours
        [TestCase(86400*2, "12:00:00 AM")] //48 hours
        public void ClockRun(int second, string expected)
        {
            for (int i = 0; i < second; i++)
            {
                _testClock.TimeIncrease();
            }
            Assert.That(_testClock.ClockDisplay(), Is.EqualTo(expected));
        }
```

```
    }
}
```

Nguyen Duc Manh

SWH02701




Test Explorer
11
11
0
Search (Ctrl+I)
Test run finished: 11 Tests (11 Passed, 0 Failed, 0 Skipped) run in 316 ms
0 Warnings
0 Errors
Test
Duration
Traits
Error Message
CounterTest2 (11) 98 ms
CounterTask (11) 98 ms
ClockT (7) 95 ms
ClockRun (4) 95 ms
ClockRun(172800,"12:00:00 A... 1 ms
ClockRun(3600,"01:00:00 AM") 94 ms
ClockRun(43200,"12:00:00 PM") < 1 ms
ClockRun(86400,"12:00:00 A... < 1 ms
InitialTime < 1 ms
TickOnce < 1 ms
TimeReset < 1 ms
CounterT (4) 3 ms
CounterInitial0 3 ms
IncrementTest < 1 ms
MultipleIncrementTest < 1 ms
ResettingTest < 1 ms
Run
Debug
Group Summary
CounterTest2
Tests in group: 11
Total Duration: 98 ms
Outcomes
11 Passed