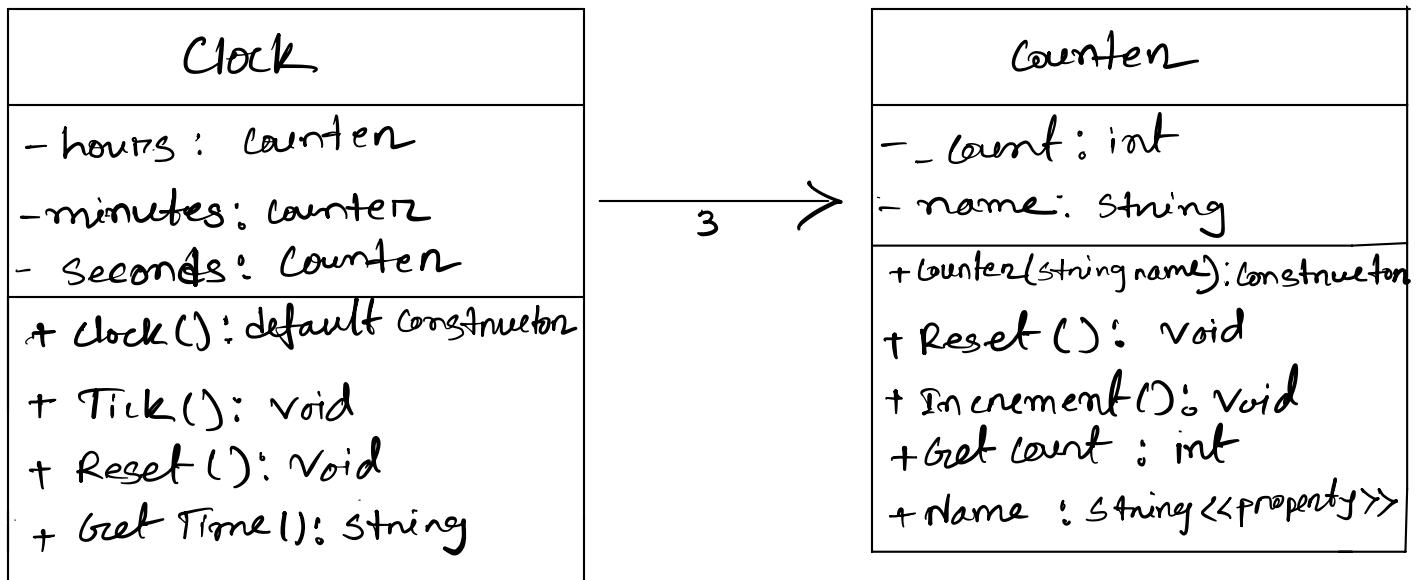


## UML Diagram - 3.1 Clock Class



```
using System;  
using ClockClass;
```

```
class Program  
{  
    static void Main(string[] args)  
    {  
        Clock clock = new Clock();  
  
        for (int i = 0; i < 100; i++)  
        {  
            clock.Tick();  
            Console.WriteLine(clock.GetTime());  
        }  
  
        clock.Reset();  
        Console.WriteLine(clock.GetTime());  
  
        Console.ReadLine();  
    }  
}
```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ClockClass
{
    public class Clock
    {
        private Counter hoursCounter;
        private Counter minutesCounter;
        private Counter secondsCounter;

        public Clock()
        {
            hoursCounter = new Counter("hours");
            minutesCounter = new Counter("minutes");
            secondsCounter = new Counter("seconds");
        }

        public void Tick()
        {
            secondsCounter.Increment();
            if (secondsCounter.Ticks == 60)
            {
                secondsCounter.Reset();
                minutesCounter.Increment();
            }
            if (minutesCounter.Ticks == 60)
            {
                minutesCounter.Reset();
                hoursCounter.Increment();
            }
            if (hoursCounter.Ticks == 24)
            {
                hoursCounter.Reset();
            }
        }

        public void Reset()
        {
            hoursCounter.Reset();
            minutesCounter.Reset();
            secondsCounter.Reset();
        }
    }
}

```

```

        public string GetTime()
    {

```

```

        {
            return $"{hoursCounter.Ticks:00}:{minutesCounter.Ticks:00}:{secondsCounter.Ticks:00}";
        }
    }
}

```

### Counter Class

```

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

```

```

public class Counter
{
    private int _count;
    private string _name;
    public Counter(string name)
    {
        _count = 0;
        _name = name;
    }
    public int Increment()
    {
        _count++;
        return _count;
    }
    public int Reset()
    {
        _count = 0;
        return _count;
    }
    public string Name
    {
        get
        {
            return _name;
        }
        set
        {
            _name = value;
        }
    }
    public int Ticks
    {
        get
        {
            return _count;
        }
    }
}

```

## Clock Test

```
using ClockClass;
```

```
namespace ClockClass;
```

```
public class ClockTests
```

```
{
```

```
    [Test]
```

```
    public void Clock_Initialises_At_Zero()
```

```
    {
```

```
        Clock clock = new Clock();
```

```
        string time = clock.GetTime();
```

```
        Assert.That(time, Is.EqualTo("00:00:00"));
```

```
    }
```

```
    [Test]
```

```
    public void Clock_Tick_Increases_Seconds_By_One()
```

```
    {
```

```
        Clock clock = new Clock();
```

```
        clock.Tick();
```

```
        string time = clock.GetTime();
```

```
        Assert.That(time, Is.EqualTo("00:00:01"));
```

```
    }
```

```
    [Test]
```

```
    public void Clock_Tick_60_Times_Increases_Minutes_By_One()
```

```
    {
```

```
        Clock clock = new Clock();
```

```
        for (int i = 0; i < 60; i++)
```

```
        {
```

```
            clock.Tick();
```

```
        }
```

```
        string time = clock.GetTime();
```

```
        Assert.That(time, Is.EqualTo("00:01:00"));
```

```
    }
```

[Test]

public void Clock\_Tick\_3600\_Times\_Increases\_Hours\_By\_One()

{

    Clock clock = new Clock();

    for (int i = 0; i < 3600; i++)

    {

        clock.Tick();

    }

    string time = clock.GetTime();

    Assert.That(time, Is.EqualTo("01:00:00"));

}

[Test]

public void Clock\_Reset\_Sets\_Time\_To\_Zero()

{

    Clock clock = new Clock();

    clock.Tick();

    clock.Tick();

    clock.Tick();

    clock.Reset();

    string time = clock.GetTime();

    Assert.That(time, Is.EqualTo("00:00:00"));

}

}

## Counter Test

```
using NUnit.Framework;
using System.Diagnostics.Metrics;

namespace ClockClass;

public class CounterTests
{
    [Test]
    public void Counter_Initialises_At_Zero()
    {
        Counter counter = new Counter("testCounter");

        int count = counter.Ticks;

        Assert.That(count, Is.EqualTo(0));
    }

    [Test]
    public void Counter_Increment_Adds_One_To_Count()
    {
        Counter counter = new Counter("testCounter");
        counter.Increment();
        int count = counter.Ticks;

        Assert.That(count, Is.EqualTo(1));
    }

    [Test]
    public void Counter_Increment_Multiple_Times_Increases_Count_To_Match()
    {
        Counter counter = new Counter("testCounter");
        counter.Increment();
        counter.Increment();
        counter.Increment();
        int count = counter.Ticks;

        Assert.That(count, Is.EqualTo(3));
    }

    [Test]
    public void Counter_Reset_Sets_Count_To_Zero()
    {
        Counter counter = new Counter("testCounter");
        counter.Increment();
        counter.Increment();
        counter.Reset();
        int count = counter.Ticks;

        Assert.That(count, Is.EqualTo(0));
    }
}
```

The screenshot displays the Visual Studio IDE with a C# project named 'ClockClass'. The code defines a `ClockClass` with a `Tick` method and a `Program` class with a `Main` method. The `Main` method creates a `Clock` object and calls `clock.Tick` in a loop. The execution window shows the program running successfully from 00:00:13 to 00:00:00.

```

using System;
using ClockClass;

class Program
{
    static void Main()
    {
        Clock clock = new ClockClass();

        for (int i = 0; i < 10; i++)
        {
            clock.Tick();
            Console.WriteLine(i);
        }
    }
}

```

Execution output:

```

00:00:13
00:00:14
00:00:15
00:00:16
00:00:17
00:00:18
00:00:19
00:00:20
00:00:21
00:00:22
00:00:23
00:00:24
00:00:25
00:00:26
00:00:27
00:00:28
00:00:29
00:00:30
00:00:31
00:00:32
00:00:33
00:00:34
00:00:35
00:00:36
00:00:37
00:00:38
00:00:39
00:00:40
00:00:00

```

Visual Studio Test Explorer showing the results of a test run for the `ClockClass` project. The test suite `ClockTests` passed successfully with a total duration of 14 ms. The test results are displayed in a table with columns for Test, Duration, Traits, and Error.

Test	Duration	Traits	Error
ClockTest (5)	14 ms		
ClockClass (5)	14 ms		
ClockTests (5)	14 ms		
Clock_Initialises_At_Zero	14 ms		
Clock_Reset_Sets_Time_To_Zero	< 1 ms		
Clock_Tick_3600_Times_Increas...	< 1 ms		
Clock_Tick_60_Times_Increases...	< 1 ms		
Clock_Tick_Increases_Seconds...	< 1 ms		
CounterTest (4)	23 ms		
ClockClass (4)	23 ms		
CounterTests (4)	23 ms		
Counter_Increment_Adds_One...	23 ms		
Counter_Increment_Multiple_Ti...	< 1 ms		
Counter_Initialises_At_Zero	< 1 ms		
Counter_Reset_Sets_Count_To...	< 1 ms		

The Group Summary on the right shows the following details:

- Group Summary**
- ClockTest**
- Tests in group: 5
- Total Duration: 14 ms
- Outcomes**
- 5 Passed