SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Counter Class

PDF generated at 17:02 on Wednesday 23rd August, 2023

```csharp
1   using System;
2
3   namespace CounterTask
4   {
5       class MainClass
6       {
7           static void PrintCounters(Counter[] counters)
8           {
9               foreach(Counter c in counters)
10              {
11                  Console.WriteLine("{0} is {1}", c.Name, c.Ticks.ToString());
12              }
13          }
14
15          public static void Main(string[] args)
16          {
17              //Declare myCounters array
18              Counter[] myCounters = new Counter[3];
19              myCounters[0] = new Counter("Counter 1");
20              myCounters[1] = new Counter("Counter 2");
21              myCounters[2] = myCounters[0];
22
23              //myCounter[0] use Increment method
24              for (int i = 0; i <= 9; i++)
25              {
26                  myCounters[0].Increment();
27              }
28
29              //myCounter[1] use Increment method
30              for (int i = 0; i <= 14 ;i++)
31              {
32                  myCounters[1].Increment();
33              }
34
35              //Print counters
36              PrintCounters(myCounters);
37
38              //Reset myCounters[2]
39              myCounters[2].Reset();
40
41              //Print counters
42              PrintCounters(myCounters);
43          }
44      }
45  }
```

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

namespace CounterTask
{
    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;
            }
        }
    }
}
```

```
28
29                //myCounter[1] use Increment method
30                for (int i = 0; i <= 14 ;i++)
31                {
32                    myCounters[1].Increment();
33                }
34
35                //Print counters
36                PrintCounters(myCounters);
37
38                //Reset myCounters[2]
39                myCounters[2].Reset();
40
41                //Print counters
42                PrintCounters(myCounters);
43            }
44        }
45    }
```

Solution
- CounterTask
  - CounterTask
    - Connected Services
    - Dependencies
    - Counter.cs
    - Program.cs

Errors

Errors: 0   Warnings: 0   Messages: 0

!	Line	Description	File	Project	Path	Category

Terminal – CounterTask

```
Counter 1 is 10
Counter 2 is 15
Counter 1 is 10
Counter 1 is 0
Counter 2 is 15
Counter 1 is 0
```