

C# WPF Coding Test - PI calculator

Please create a WPF application using Visual Studio and use MVVM design pattern to complete the task.

1. Calculate PI by simulation

Create a mission class containing a member function that calculates PI by simulation.

Please note that you are expected to apply multi-threading / parallel computing methods to accelerate to procedure. Please try your best to improve performance of the Calculate function.

```
public class PiMission
{
    public readonly long SampleSize

    public double Calculate() {
        // please calculate PI by the given sample size
        // return the calculated value in this function
    }

    public PiMission(long sampleSize)
    {
        this.SampleSize = sampleSize;
    }
}
```

2. Create GUI

Build a window that displays the results

PI Calculator		
Sample Size	100000	Add
Sample	Time	Value
200000	16:45:00	3.141592655
300000	16:45:01	3.14162522

- The DataGrid contains three columns
 - Sample: displays the sample size of the mission
 - Time: displays the latest calculation time in the format "HH:mm:ss"
 - Value: displays the latest calculated PI
- Each row represents a PiMission with a specific sample size
- When user clicks the "Add" button, it adds a mission that calculates PI with the sample size indicated by the textbox if the existing mission list does not have a mission using the size. i.e.
 - enter 10000 and click "Add" will add an PiMission to the list with sample size 10000
 - enter 20000 and click "Add" will add an PiMission to the list with sample size = 20000
 - But if now we enter 10000 again and click "Add", it should not add another PiMission with sample size = 10000 to the list.
- The window should automatically refresh every second.