Test Driven Development and the V-Model

Sebastian Venginickal

November 27, 2007

ENPM 643

Introduction

- What is Test Driven Development?
- Where does it fit into the V-Model?
- How can it help us?
- How do I use TDD?
- A practical example from my work
  - Multiprobe spectroscopy with a shared detector.

What is TDD?

- MORE than just a “testing” method
- Software design and implementation technique
  - Covers Requirements -> Test
- Many tenants and aspects, but key is:
    - WRITE TEST FIRST.

Why Write the Tests First?

- Forces developers to consider "**what**" each module must do.
  - The **Requirements** before the production code
- Forces the developer to consider "**how**" the module will be used
  - The **Interface Design** before the production code
- Encourages cohesive modules with little coupling

What is a Test?

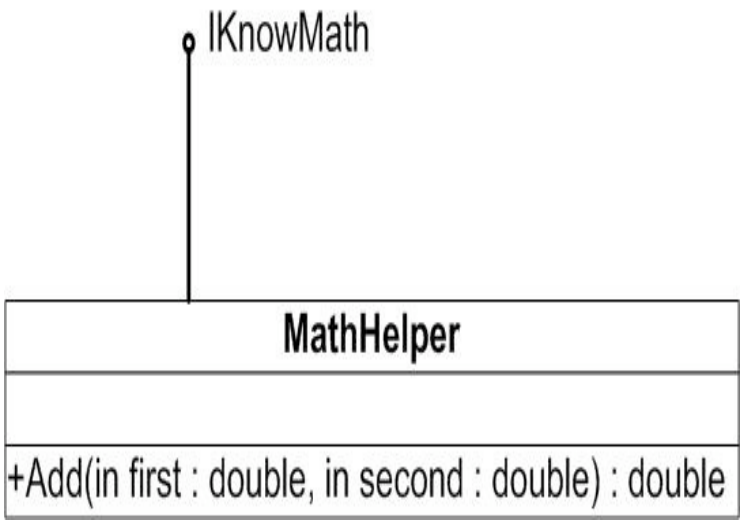
- Code written with the specific and only purpose to test another modules
- Test defines the requirements of the module being tested
- These tests contain assertions that are either true or false. (i.e. Pass or Fail)
- Test can be automated
  - run fast and get immediate results

Extremely Simple Test

- Simple “MathHelper” that implements IKnowMath interface

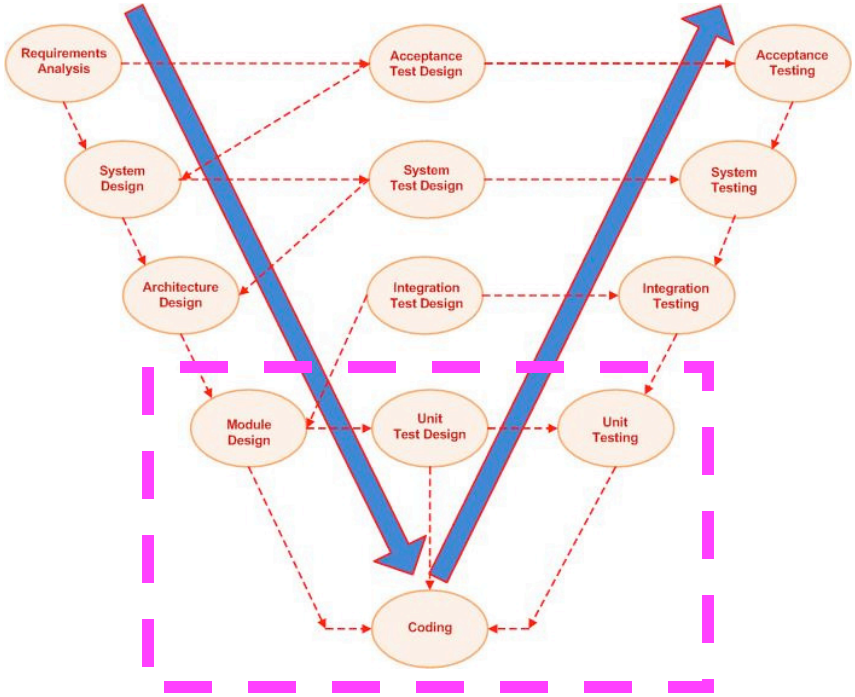
```
double sum = IKnowMath.Add(3, 5);

If (sum == 8)
        return true; //PASS ASSERT
else
        return false; //FAIL ASSERT
```



What about the V-Model?

- Fits in nicely to the bottom of the V
- TDD emphasizes a design approach that focuses on creating testable modules
- Do your test design, and module design, BEFORE the code.



How does TDD help?

- More likely to write good production code the first time. (i.e. tests will pass)
- Most code can be tested before it ever gets into production.
- Code changes, **refactorings**, etc. can be quickly validated.
- Process leads to simpler, cleaner designs

Where does it Not help?

- Not a perfect catch all technique
- Bad tests lead to bad code
- Difficult for graphical user interfaces
  - Too many input variations to predict and test
- Difficult for Relational Database applications
  - Too many interdependencies
- Lots of developer time spent writing Non-production code

How do I use TDD?

- 0. Know the Requirement (Feature)
- 1. Write a Test for the feature
  - It will fail the first time
- 2. Implement the feature quickly
  - Just good enough to make the test pass
- 3. Refactor the code
  - Make the implementation cleaner, quicker, more efficient, etc.
  - All Tests must still pass

Advanced Concepts

- Never as simple as “Add”
- Tests should never access anything “external” to the module being tested
  - Other modules, data stores, processes, internet, etc.
  - Test and modules should be cohesive units
  - Hard to do

More Advanced Concepts

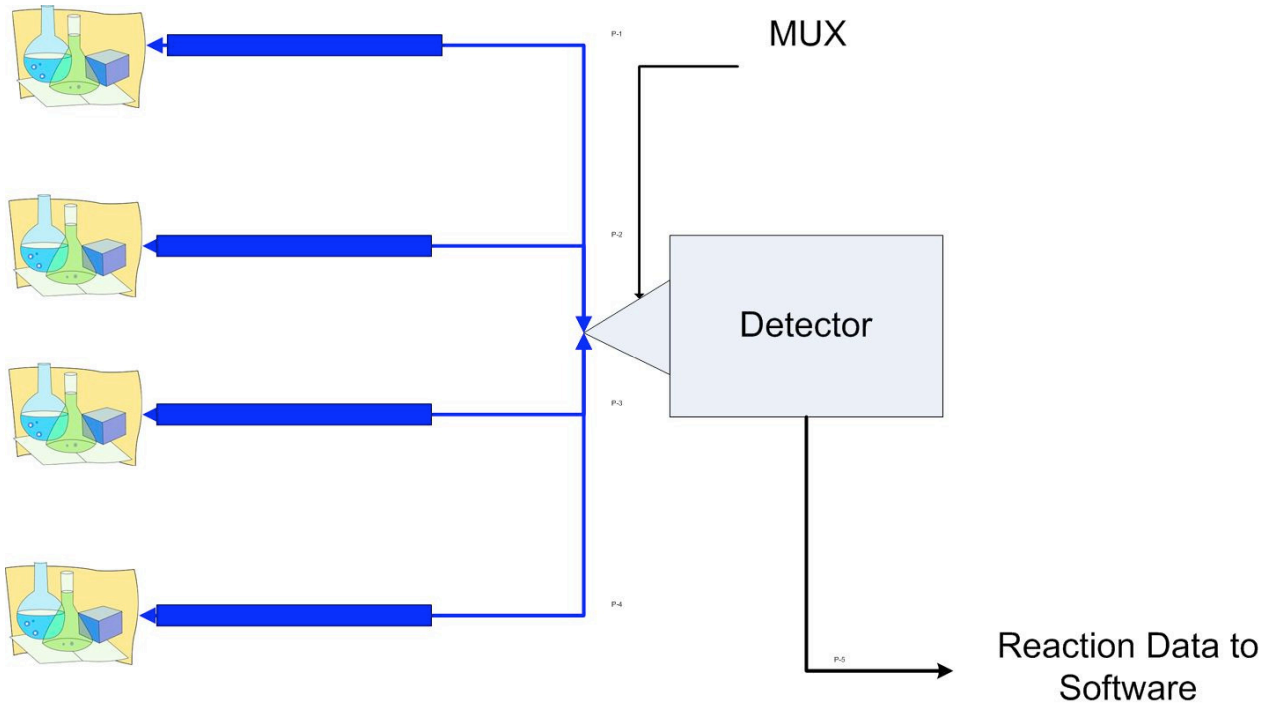
- Design by Contract
  - Focus on Interface design, “How Modules Communicate”, for external needs
- Mock Objects
  - Implement the interface and validate input or output data
- Fake Objects
  - Simply implement the interface and return a success.

Practical Example

- Real project I have for Q1 2008
- Show simple use for TDD
- Mettler Toledo Autochem
  - Instrumentation for Chemical and Pharmaceutical Research
- Chemists use our instrument for analysis of reactions and product development

The Problem

- Multiprobe data collection with a single detector



Use Case

Use Case Name:	StartExperiment	ID:	1.0	Importance Level:	High
Primary Actor: Chemist			Use Case Type: Association		
Stakeholders and Interests: Chemist					
Brief Description: **Chemist starts two different experiments in two separate reactors at the same time**					
Trigger: Chemist starts the experiments Type: Overview					
…					
…					
Subflows:					
Alternate/ exceptional Flows: 3) The Chemist selects 1, 3, or 4 probes. 5-6) The chemist calibrates each selected probe					

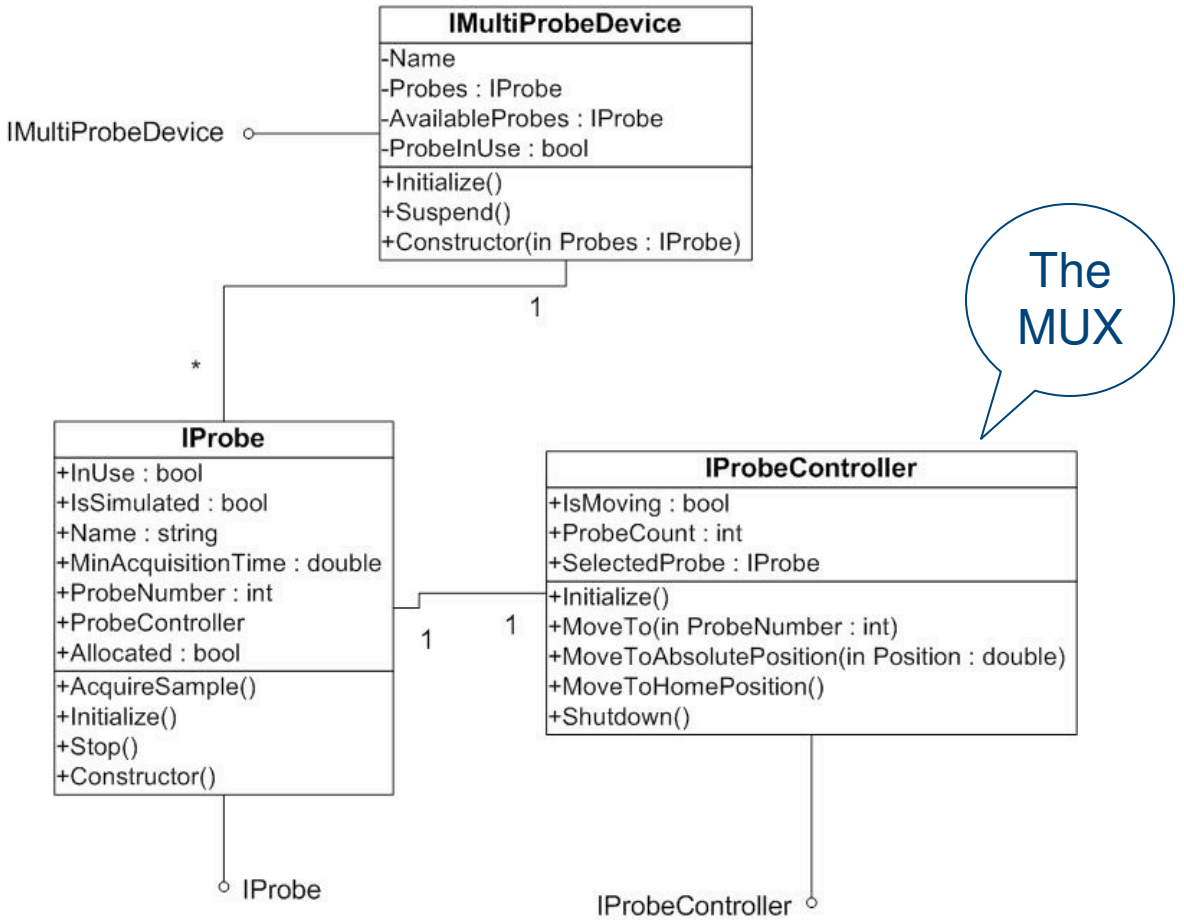
Use Case: Normal Flow of Events

1) The Chemist gathers material for both experiments
2) The Chemist starts the software and clicks New Experiment
3) The Chemist selects the probes for the experiment
4) The Chemist sets the sample interval for each experiment
5) The Chemist calibrates the first probe
6) The Chemist calibrates the second probe
6) The Chemist loads the materials into the reactors
7) The Chemist puts the probes in the reactor
8) The Chemist starts the experiment and begins collecting data from both vessels at the specified interval.

Requirements

- The system shall allow the chemist to choose which probes to use in an experiment.
- *A probe shall be used for only one experiment.*
- The instrument shall collect data at the user prescribed interval.
- The instrument shall collect data from multiple probes for multiple, simultaneous experiments
- *The instrument shall only collect from 1 probe at any instant time*

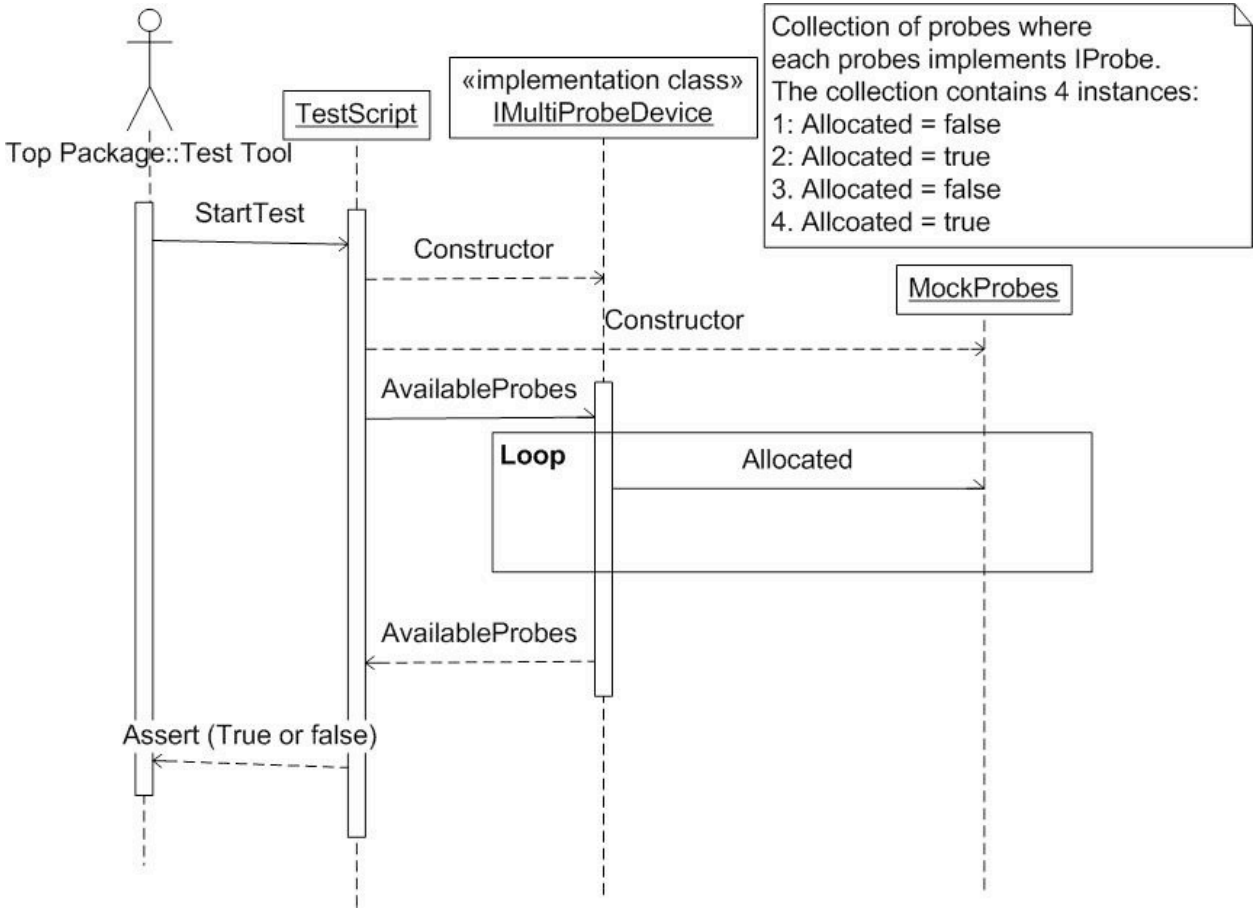
Interface and Class Design

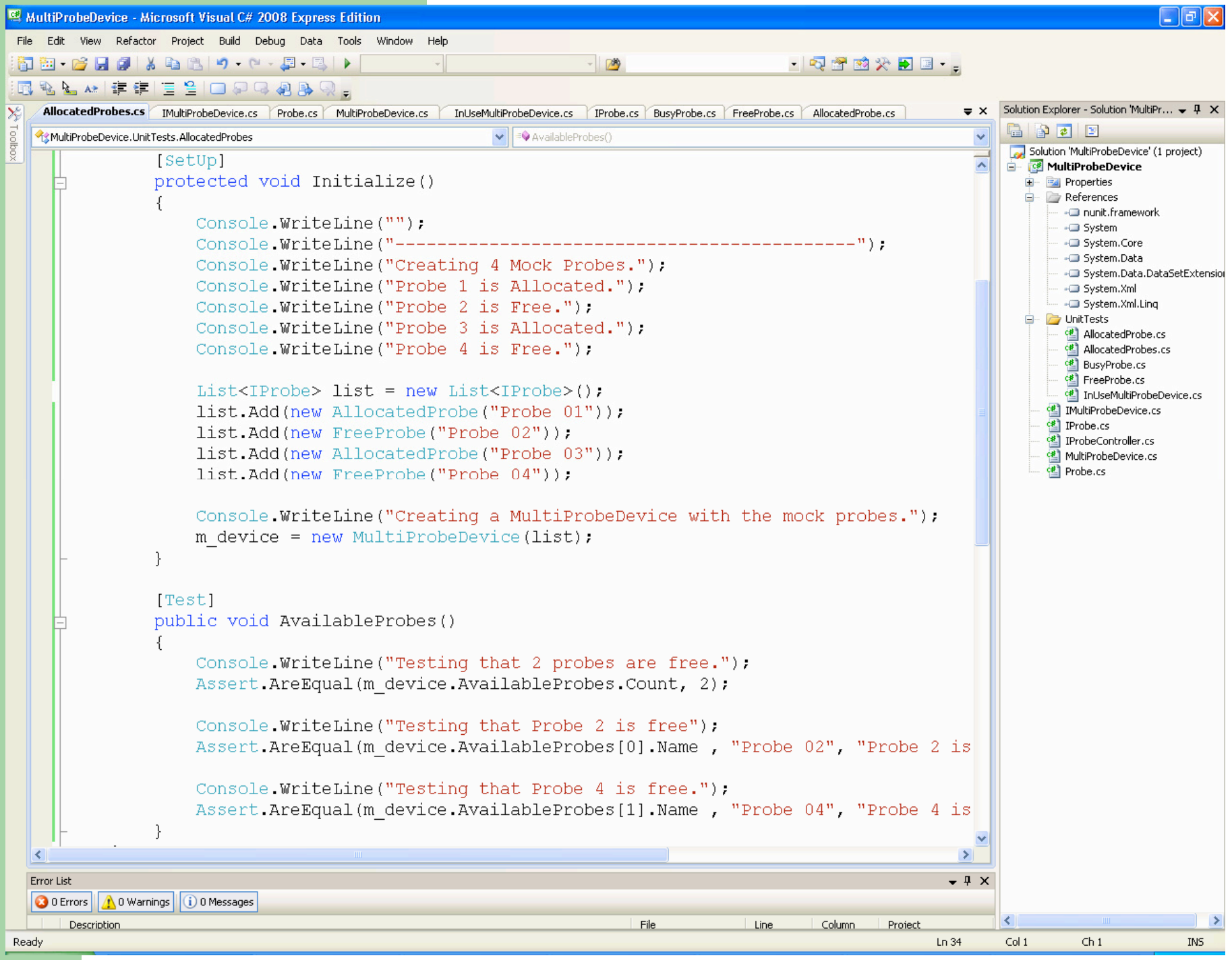


The Tests

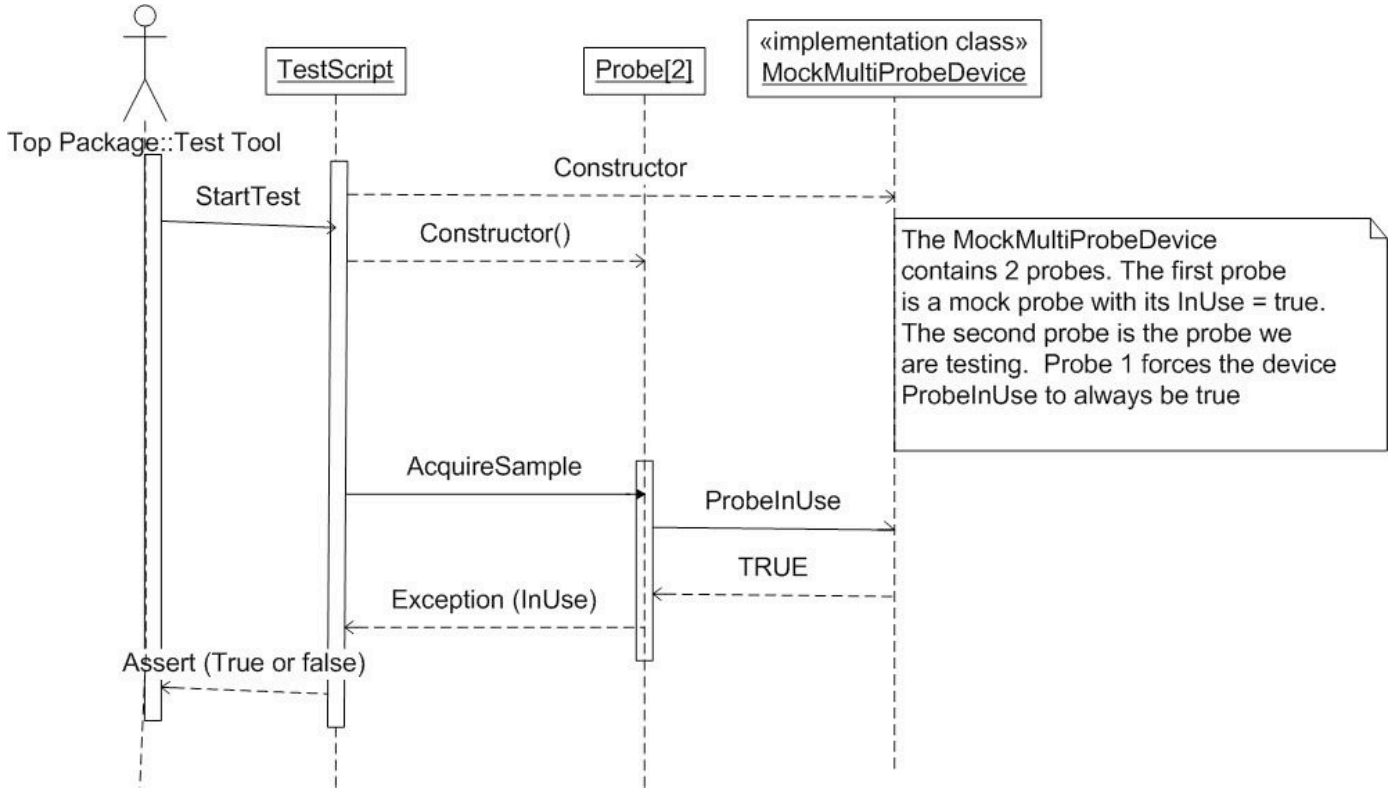
- A probe can be used for only one experiment.
  - [IMultiProbeDevice.AvailableProbes]
- The instrument shall only collect from 1 probe at any instant time
  - [IProbe.InUse]

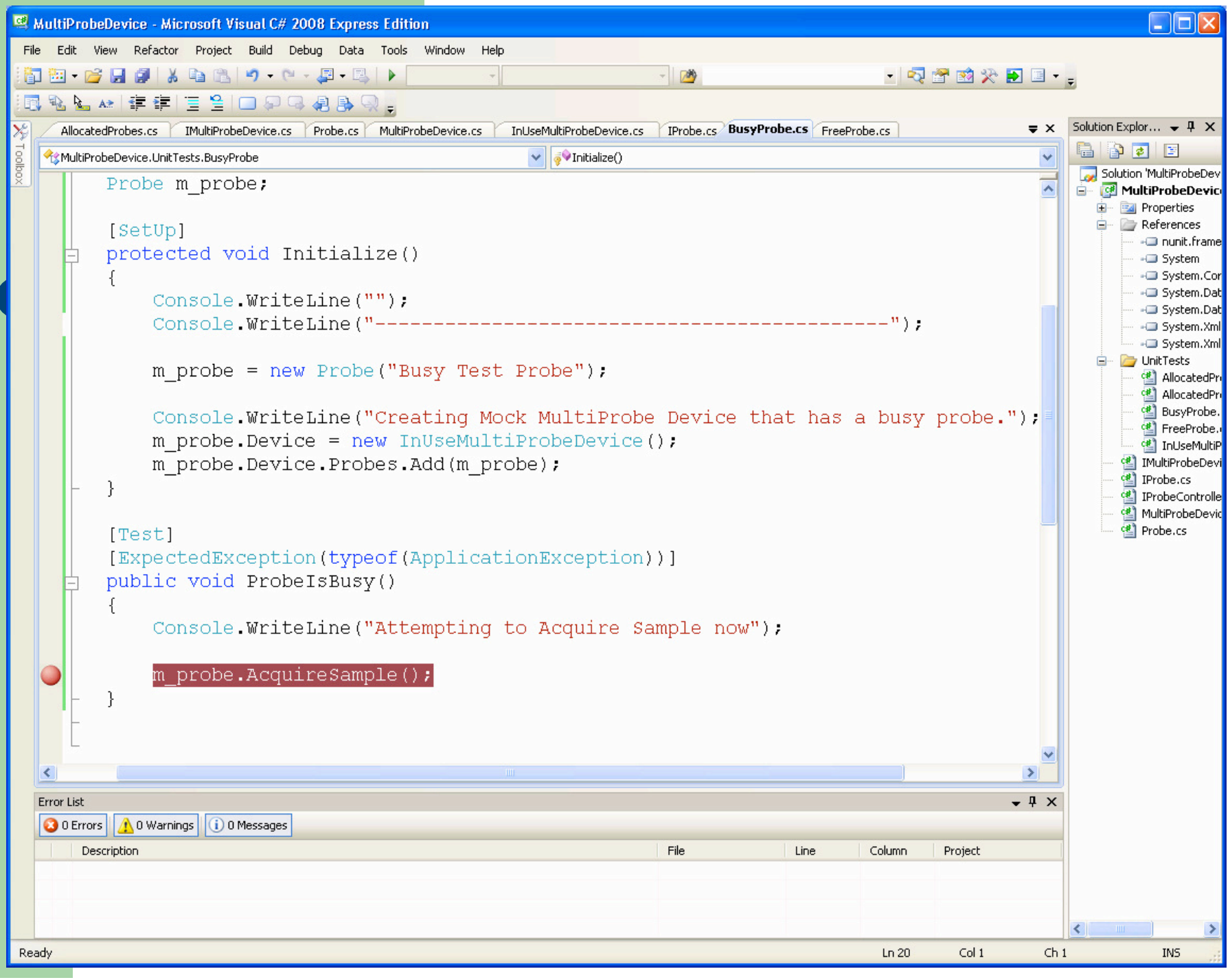
The Tests: Available Probes



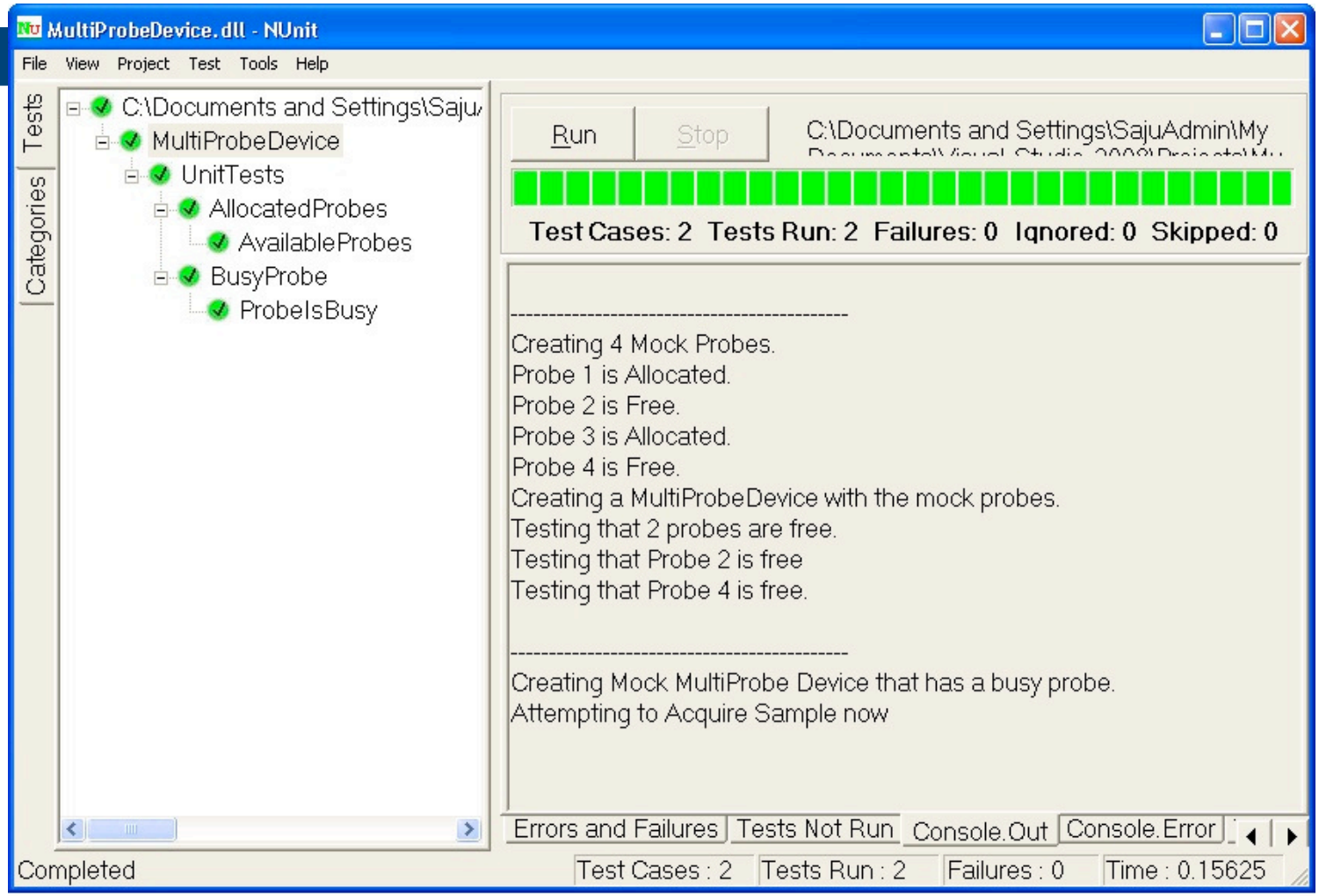

MultiProbeDevice - Microsoft Visual C# 2008 Express Edition
File Edit View Refactor Project Build Debug Data Tools Window Help
AllocatedProbes.cs IMultiProbeDevice.cs Probe.cs MultiProbeDevice.cs InUseMultiProbeDevice.cs IProbe.cs BusyProbe.cs FreeProbe.cs AllocatedProbe.cs
MultiProbeDevice.UnitTests.AllocatedProbes
AvailableProbes()
[SetUp]
protected void Initialize()
{
Console.WriteLine("");
Console.WriteLine("----------");
Console.WriteLine("Creating 4 Mock Probes.");
Console.WriteLine("Probe 1 is Allocated.");
Console.WriteLine("Probe 2 is Free.");
Console.WriteLine("Probe 3 is Allocated.");
Console.WriteLine("Probe 4 is Free.");
List<IProbe> list = new List<IProbe>();
list.Add(new AllocatedProbe("Probe 01"));
list.Add(new FreeProbe("Probe 02"));
list.Add(new AllocatedProbe("Probe 03"));
list.Add(new FreeProbe("Probe 04"));
Console.WriteLine("Creating a MultiProbeDevice with the mock probes.");
m_device = new MultiProbeDevice(list);
}
[Test]
public void AvailableProbes()
{
Console.WriteLine("Testing that 2 probes are free.");
Assert.AreEqual(m_device.AvailableProbes.Count, 2);
Console.WriteLine("Testing that Probe 2 is free");
Assert.AreEqual(m_device.AvailableProbes[0].Name , "Probe 02", "Probe 2 is
Console.WriteLine("Testing that Probe 4 is free.");
Assert.AreEqual(m_device.AvailableProbes[1].Name , "Probe 04", "Probe 4 is
}
Solution Explorer - Solution 'MultiPr...
Solution 'MultiProbeDevice' (1 project)
MultiProbeDevice
Properties
References
nunit.framework
System
System.Core
System.Data
System.Data.DataSetExtensio
System.Xml
System.Xml.Linq
UnitTests
AllocatedProbe.cs
AllocatedProbes.cs
BusyProbe.cs
FreeProbe.cs
InUseMultiProbeDevice.cs
IMultiProbeDevice.cs
IProbe.cs
IProbeController.cs
MultiProbeDevice.cs
Probe.cs
Error List
0 Errors 0 Warnings 0 Messages
Description File Line Column Project
Ready
Ln 34 Col 1 Ch 1 INS

The Test: In Use




MultiProbeDevice - Microsoft Visual C# 2008 Express Edition
File Edit View Refactor Project Build Debug Data Tools Window Help
AllocatedProbes.cs IMultiProbeDevice.cs Probe.cs MultiProbeDevice.cs InUseMultiProbeDevice.cs IProbe.cs BusyProbe.cs FreeProbe.cs
MultiProbeDevice.UnitTests.BusyProbe
Initialize()
Probe m_probe;
[SetUp]
protected void Initialize()
{
Console.WriteLine("");
Console.WriteLine("----------");
m_probe = new Probe("Busy Test Probe");
Console.WriteLine("Creating Mock MultiProbe Device that has a busy probe.");
m_probe.Device = new InUseMultiProbeDevice();
m_probe.Device.Probes.Add(m_probe);
}
[Test]
[ExpectedException(typeof(ApplicationException))]
public void ProbeIsBusy()
{
Console.WriteLine("Attempting to Acquire Sample now");
m_probe.AcquireSample();
}
Solution Explor...
Solution 'MultiProbeDev
MultiProbeDevic
Properties
References
UnitTests
IMultiProbeDevi
IProbe.cs
IProbeControlle
MultiProbeDevic
Probe.cs
Error List
0 Errors 0 Warnings 0 Messages
Description File Line Column Project
Ready Ln 20 Col 1 Ch 1 INS

Test Results



Conclusion

- Defined Test Driven Development
- Showed some TDD Techniques for design and test
- Demonstrated it with simple example from my work
- Briefly showed where it fits into V-Model

Questions?