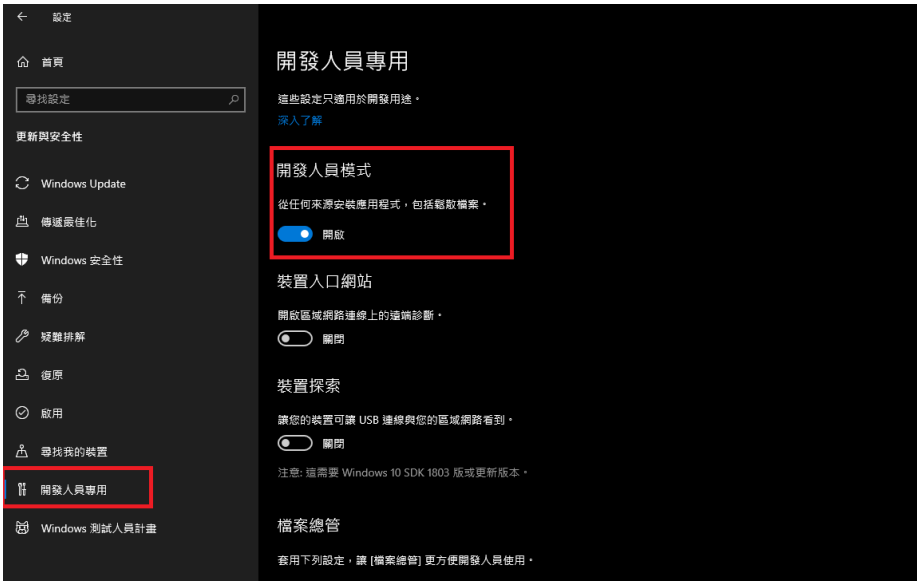
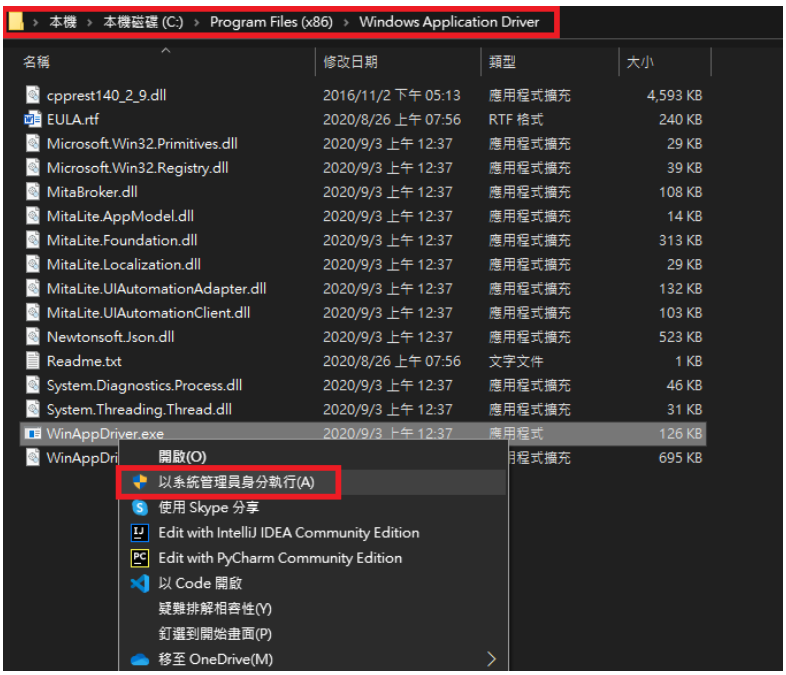
UI 測試教學

一、 環境設定

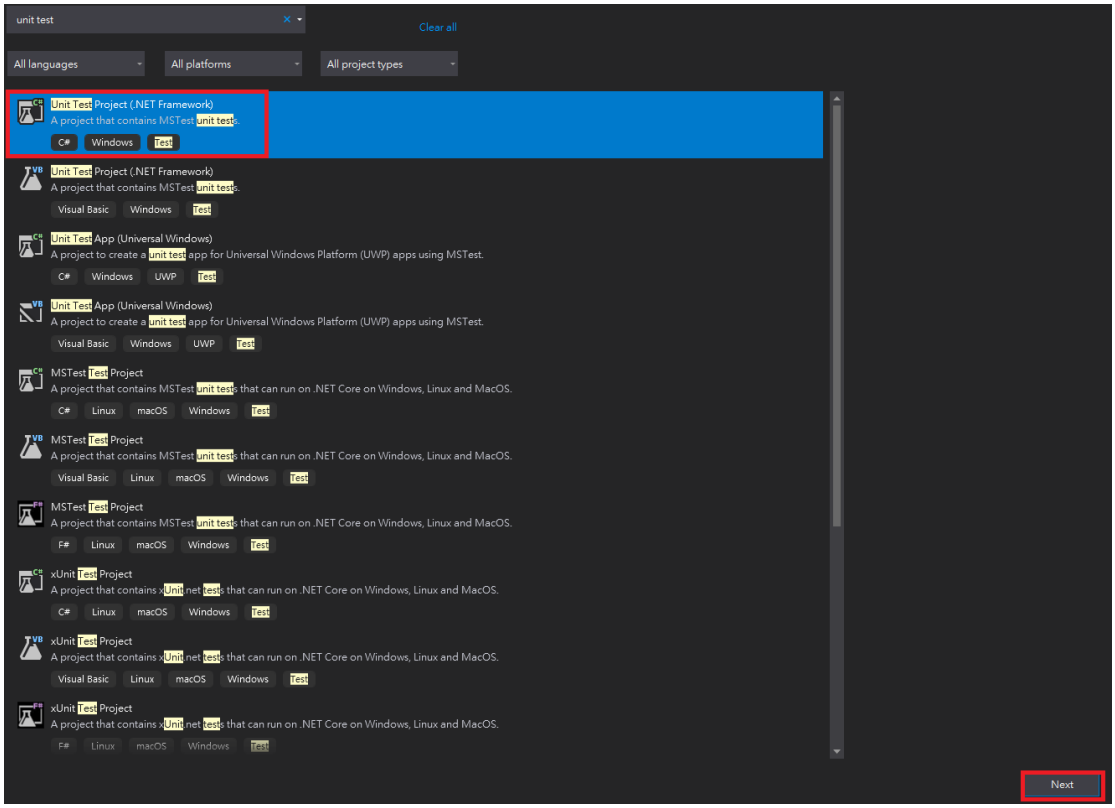
1. 開啟 Windows 開發人員模式
   - 設定 -> 更新與安全性 -> 開發人員專用 -> 開發人員模式



2. Winappdriver
   - 下載 winappdriver 並安裝
   - 進行測試前，請先以系統管理員身分開啟 winappdriver (預設路徑為: *C:\Program Files (x86)\Windows Application Driver\WinAppDriver.exe*)

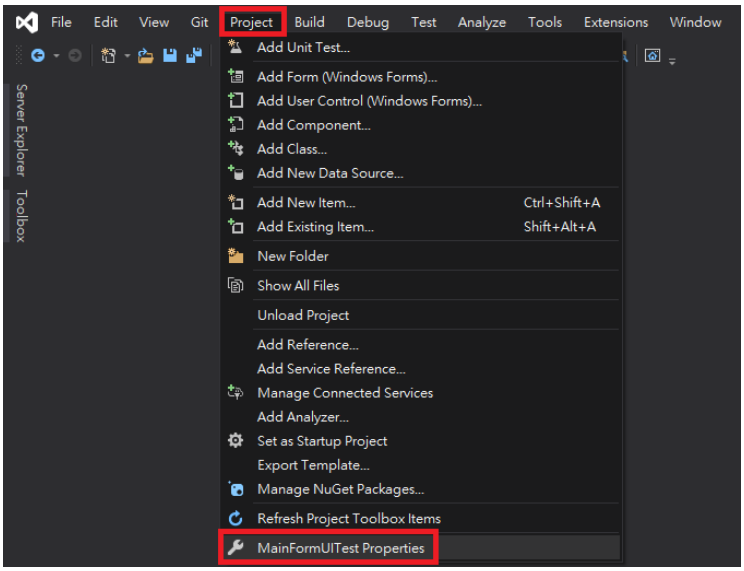

本機 › 本機磁碟 (C:) › Program Files (x86) › Windows Application Driver
名稱
修改日期
類型
大小
cpprest140_2_9.dll 2016/11/2 下午 05:13 應用程式擴充 4,593 KB
EULA.rtf 2020/8/26 上午 07:56 RTF 格式 240 KB
Microsoft.Win32.Primitives.dll 2020/9/3 上午 12:37 應用程式擴充 29 KB
Microsoft.Win32.Registry.dll 2020/9/3 上午 12:37 應用程式擴充 39 KB
MitaBroker.dll 2020/9/3 上午 12:37 應用程式擴充 108 KB
MitaLite.AppModel.dll 2020/9/3 上午 12:37 應用程式擴充 14 KB
MitaLite.Foundation.dll 2020/9/3 上午 12:37 應用程式擴充 313 KB
MitaLite.Localization.dll 2020/9/3 上午 12:37 應用程式擴充 29 KB
MitaLite.UIAutomationAdapter.dll 2020/9/3 上午 12:37 應用程式擴充 132 KB
MitaLite.UIAutomationClient.dll 2020/9/3 上午 12:37 應用程式擴充 103 KB
Newtonsoft.Json.dll 2020/9/3 上午 12:37 應用程式擴充 523 KB
Readme.txt 2020/8/26 上午 07:56 文字文件 1 KB
System.Diagnostics.Process.dll 2020/9/3 上午 12:37 應用程式擴充 46 KB
System.Threading.Thread.dll 2020/9/3 上午 12:37 應用程式擴充 31 KB
WinAppDriver.exe 2020/9/3 上午 12:37 應用程式 126 KB
WinAppDri 695 KB
開啟(O)
以系統管理員身分執行(A)
使用 Skype 分享
Edit with IntelliJ IDEA Community Edition
Edit with PyCharm Community Edition
以 Code 開啟
疑難排解相容性(Y)
釘選到開始畫面(P)
移至 OneDrive(M)

二、 建立測試專案

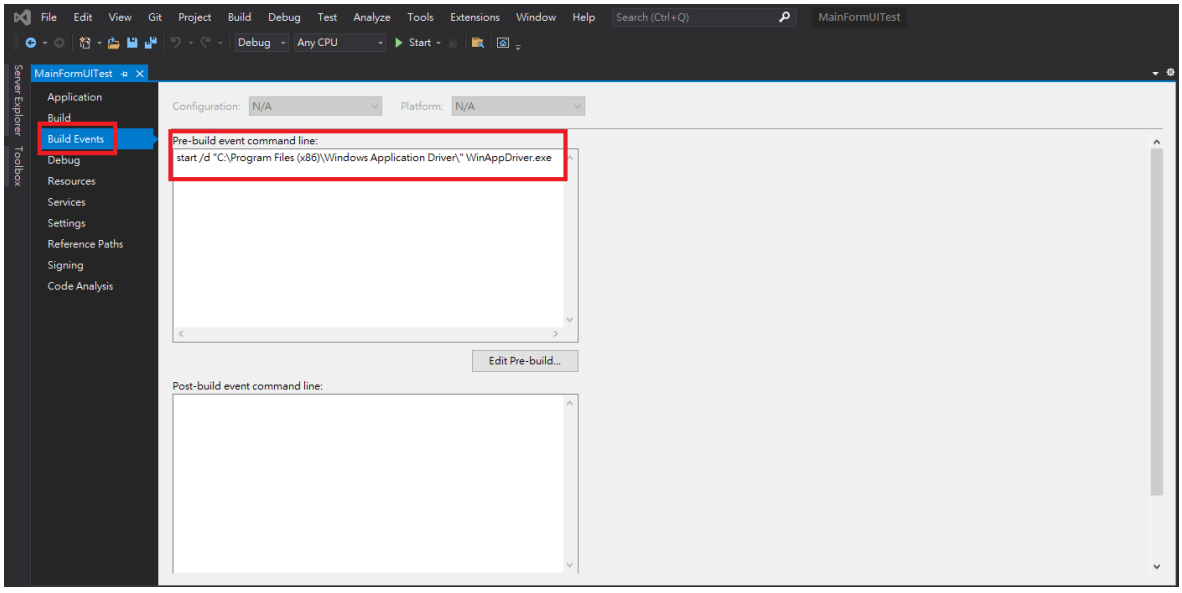

unit test
Clear all
All languages
All platforms
All project types
Unit Test Project (.NET Framework)
A project that contains MSTest unit tests
C# Windows Test
Unit Test Project (.NET Framework)
A project that contains MSTest unit tests
Visual Basic Windows Test
Unit Test App (Universal Windows)
A project to create a unit test app for Universal Windows Platform (UWP) apps using MSTest.
C# Windows UWP Test
Unit Test App (Universal Windows)
A project to create a unit test app for Universal Windows Platform (UWP) apps using MSTest.
Visual Basic Windows UWP Test
MSTest Test Project
A project that contains MSTest unit tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Test
MSTest Test Project
A project that contains MSTest unit tests that can run on .NET Core on Windows, Linux and MacOS.
Visual Basic Linux macOS Windows Test
MSTest Test Project
A project that contains MSTest unit tests that can run on .NET Core on Windows, Linux and MacOS.
F# Linux macOS Windows Test
xUnit Test Project
A project that contains xUnit.net tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Test
xUnit Test Project
A project that contains xUnit.net tests that can run on .NET Core on Windows, Linux and MacOS.
Visual Basic Linux macOS Windows Test
xUnit Test Project
A project that contains xUnit.net tests that can run on .NET Core on Windows, Linux and MacOS.
F# Linux macOS Windows Test
Next

三、 設定 VS pre-build

- 開啟 VS2019
- project -> project properties -> Build Event -> Pre-build event command line

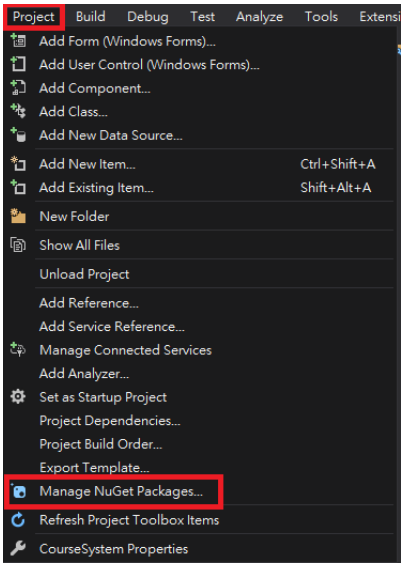


- 輸入 *start /elevate /d "C:\Program Files (x86)\Windows Application Driver\" WinAppDriver.exe* 後，每次執行測試前將會自動開啟 winappdriver

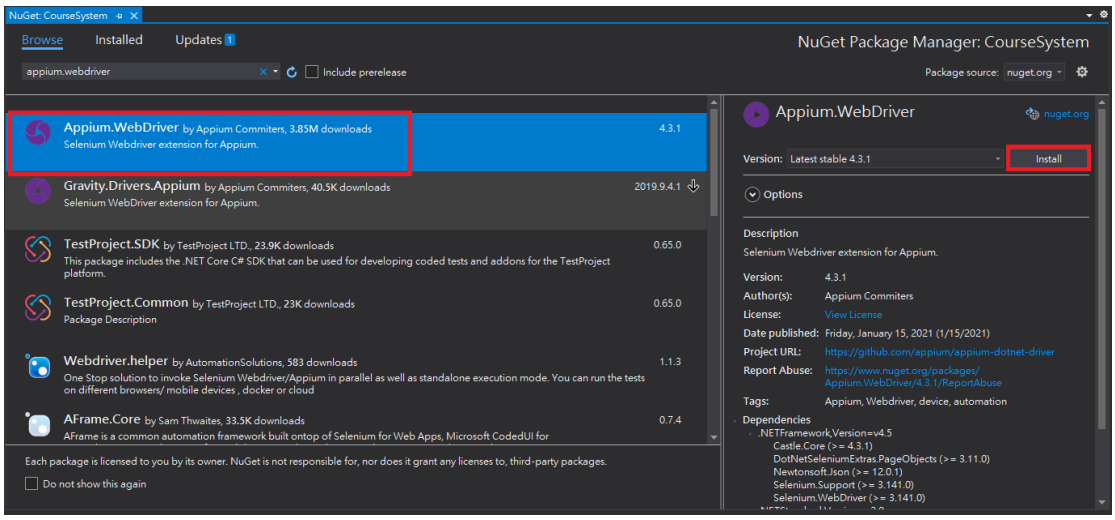


四、 安裝 Appium

- project -> Manage NuGet Packages…

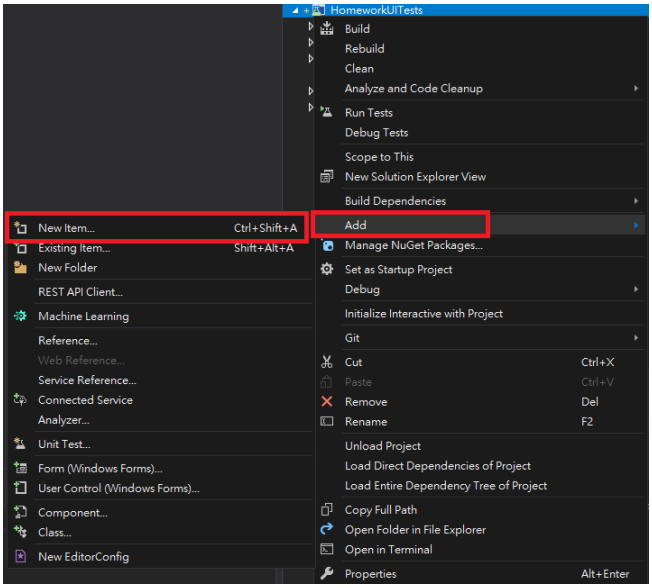


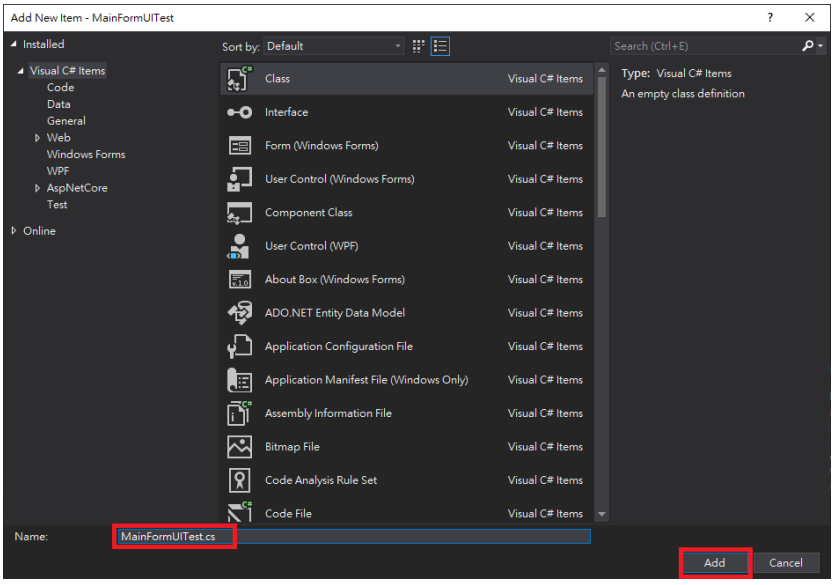
- install appium.webdriver



五、 加入測試檔案

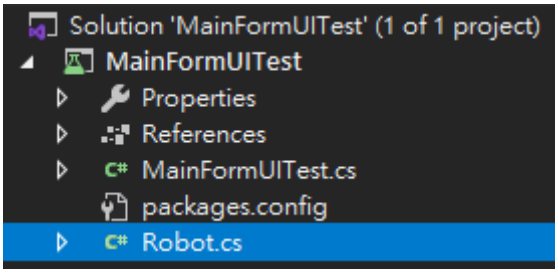
- add -> new item -> class



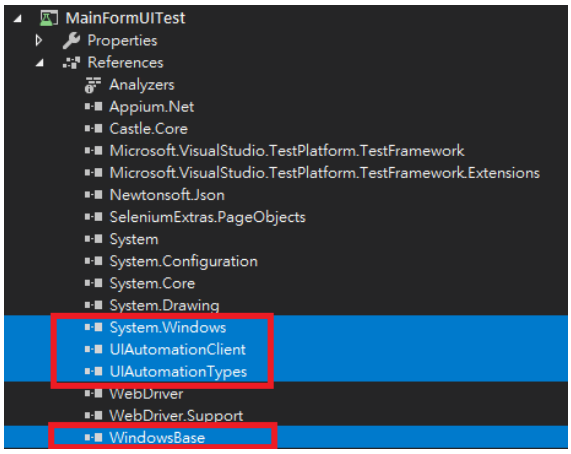


六、 引入 Robot.cs

- 確認 Robot.cs 的 namespace 是一樣的



- 加入以下參考: ***System.Windows, UIAutomationClient, UIAutomationTypes, WindowsBase***



七、 修改 MainFormUITest.cs

```
/// <summary>
/// Summary description for MainFormUITest
/// </summary>
[TestClass()]
public class MainFormUITest
{
	private Robot _robot;
	private const string APP_NAME =
"Microsoft.WindowsCalculator_8wekyb3d8bbwe!App";
	private const string CALCULATOR_TITLE = "小算盤";
	private const string EXPECTED_VALUE = "顯示是 444";
	private const string RESULT_CONTROL_NAME = "CalculatorResults";

	/// <summary>
	/// Launches the Calculator
	/// </summary>
	[TestInitialize()]
	public void Initialize()
	{

	}

	/// <summary>
	/// Closes the launched program
	/// </summary>
	[TestCleanup()]
	public void Cleanup()
	{

	}
}
```

八、 開啟小算盤

```
/// <summary>
/// Launches the Calculator
/// </summary>
[TestInitialize()]
public void Initialize()
{
      _robot = new Robot(APP_NAME, CALCULATOR_TITLE);
}
```

九、 關閉小算盤

```
/// <summary>
/// Closes the launched program
/// </summary>
[TestCleanup()]
public void Cleanup()
{
      _robot.CleanUp();
}
```

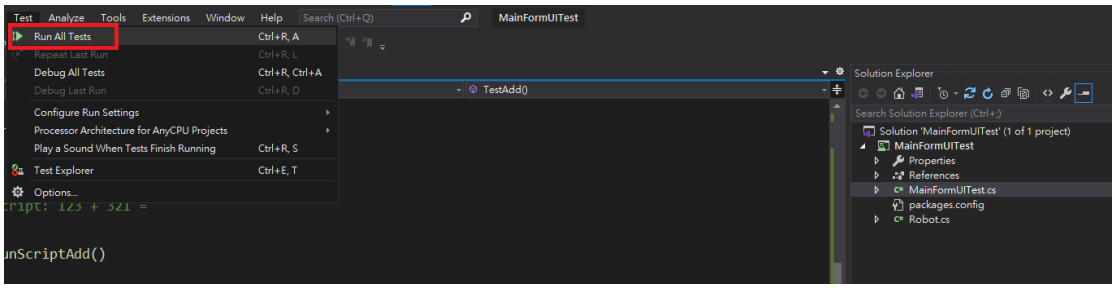
十、 撰寫點擊按鈕的程式

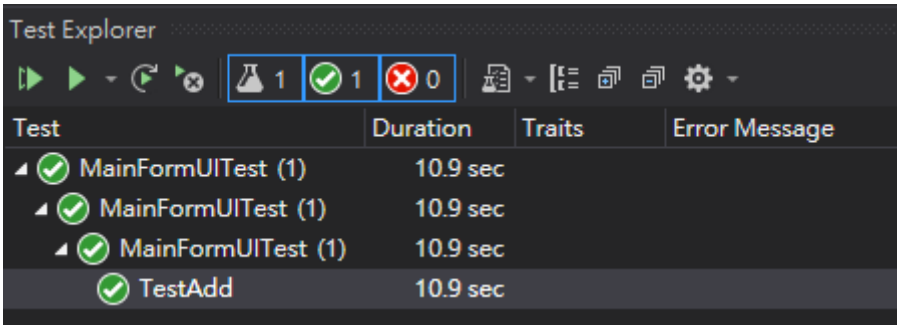
```
/// <summary>
/// Runs the script: 123 + 321 =
/// </summary>
private void RunScriptAdd()
{
        _robot.ClickButton("清除");
        _robot.ClickButton("一");
        _robot.ClickButton("二");
        _robot.ClickButton("三");
        _robot.ClickButton("加");
        _robot.ClickButton("三");
        _robot.ClickButton("二");
        _robot.ClickButton("一");
        _robot.ClickButton("等於");
}
```

十一、 確認結果

```
/// <summary>
/// Tests that the result of 123 + 321 should be 444
/// </summary>
[TestMethod]
public void TestAdd()
{
        RunScriptAdd();
        _robot.AssertText(RESULT_CONTROL_NAME, EXPECTED_VALUE);
}
```

十二、 執行





十三、 Appendix

1. 測試其他程式
   - 如果要測試其他的程式，只要將程式的檔案路徑傳入 Robot.Initialize 中即可
   - 以下以作業程式作為範例，取得程式路徑程式碼

```
private string targetAppPath;
private const string START_UP_FORM = "StartUpForm";


// init
[TestInitialize]
public void Initialize()
{
      var projectName = "CourseSystem";
      string solutionPath = Path.GetFullPath(Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"..\\..\\..\\"));
      targetAppPath = Path.Combine(solutionPath, projectName, "bin", "Debug", "CourseSystem.exe");
      _robot = new Robot(targetAppPath, START_UP_FORM);
}
```

2. 利用 Windows SDK inspect.exe 協助測試

- 安裝 VS 時，就會安裝 Windows SDK，因此不須額外安裝，如果要安裝其他版本的 Windows SDK，可使用 VS installer

- inspect 程式路徑 *C:\Program Files (x86)\Windows Kits\10\bin\{ SDK 版本 }\x64\inspect.exe*
- 執行程式時，可使用 inspect 來查看元件名稱或結構
- 以下圖為例，當滑鼠點擊 DataGridView 中的視窗程式設計課程時，DataGridView 的結構會顯示在左方，而欄位的資訊顯示在右方。