

BÀI TẬP THỰC HÀNH 1

KIỂM THỬ ĐƠN VỊ VỚI NUNIT

1. Chuẩn đầu ra cần đạt

Bài tập này nhằm mục tiêu đạt được các chuẩn đầu ra sau:

- Áp dụng phương pháp kiểm thử hộp trắng để thiết kế ca kiểm thử.
- Viết được kịch bản và chạy kiểm thử với NUnit

2. Hướng dẫn sử dụng NUnit 2.6.4

2.1. Giới thiệu unit testing

Sau khi viết code xong, lập trình viên phải thực hiện việc kiểm tra xem code của mình viết có đúng không. Công việc này gọi là unit test. Một số thông tin về unit test:

- Unit test được thực hiện bởi lập trình viên (programmer)
- Bộ phận QA sẽ thực hiện test nhưng không phải là unit test
- Mọi lập trình viên đều phải thực hiện & báo cáo kết quả unit test trước, trong hoặc sau khi develop.
- Unit test có thể thực hiện tự động sử dụng các đoạn code gọi là test script. Quá trình này gọi là automated unit testing
- Trong automated unit testing, ta sẽ viết các **hàm kiểm tra** gọi các **hàm cần kiểm tra**. Nếu kết quả trả ra như mong đợi thì kết quả là Pass ngược lại là Failed. Sau khi thực thi tất cả các hàm kiểm tra ta sẽ có được 1 bảng tóm tắt kết quả: bao nhiêu Pass, bao nhiêu Failed và các lý do.
- Automated unit test script có thể được lưu lại và chạy lại 1 cách dễ dàng khi mã nguồn thay đổi
- *Nunit là 1 framework giúp chúng ta thực hiện unit testing trên .NET. Ngoài Nunit ta còn có Junit cho java, cppunit cho cpp, phpunit for php ...*

2.2. Hướng dẫn cài đặt

- Tải Nunit bản 2.6.4 tại địa chỉ: <http://www.nunit.org>
- Sau khi tải, nhấn đúp để bắt đầu cài đặt. Làm theo các hướng dẫn. Có thể chọn Setup type là Typical hay Complete.

Nunit

HOMEDOWNLOADDOCUMENTATIONWIKIBLOGCONTACT US

Downloads

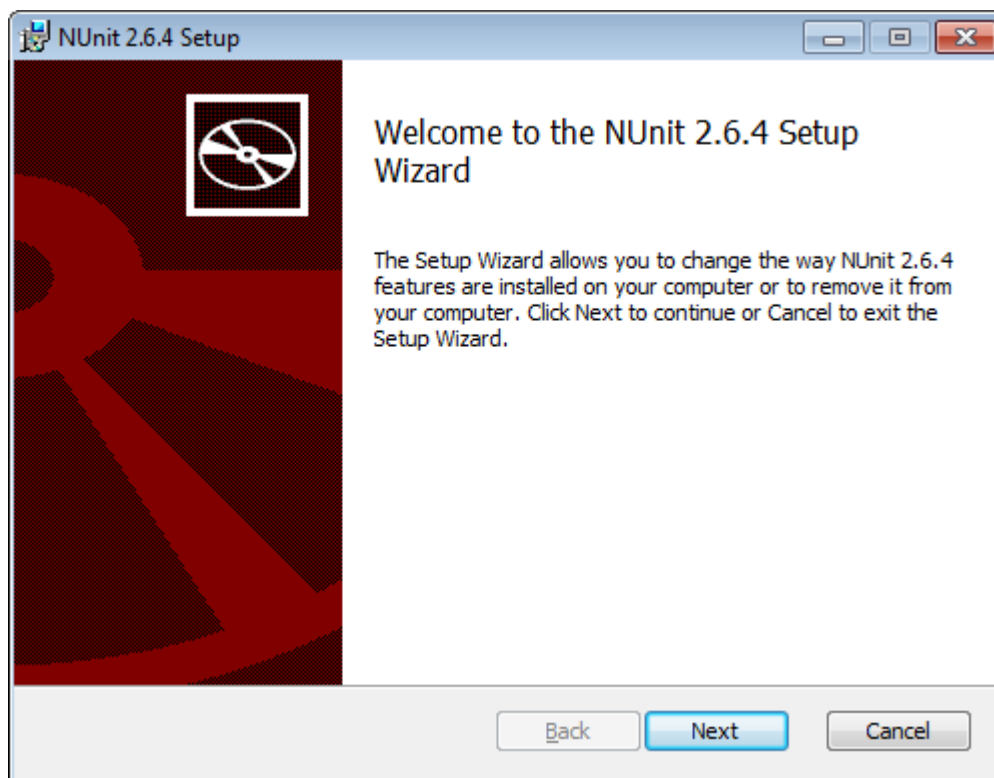
Current Stable Release: NUnit 2.6.4

NUnit 2.6.4 - December 16, 2014	
win	NUnit-2.6.4.msi
bin	NUnit-2.6.4.zip
win .net 1.1	NUnit-2.6.4-net-1.1.msi
bin .net 1.1	NUnit-2.6.4-net-1.1.zip
src	(currently unavailable)
doc	NUnit-2.6.4-docs.zip

Under Development: NUnit 3.0

NUnit 3.0 Beta 4 - August 25, 2015	
win	NUnit.3.0.0-beta-4.msi
bin	NUnit-3.0.0-beta-4.zip
src	3.0.0-beta-4-src.zip
cf	NUnitCF-3.0.0-beta-4.zip

Additional downloads for NUnit 3.0 can be found on the [GitHub releases pages](#).

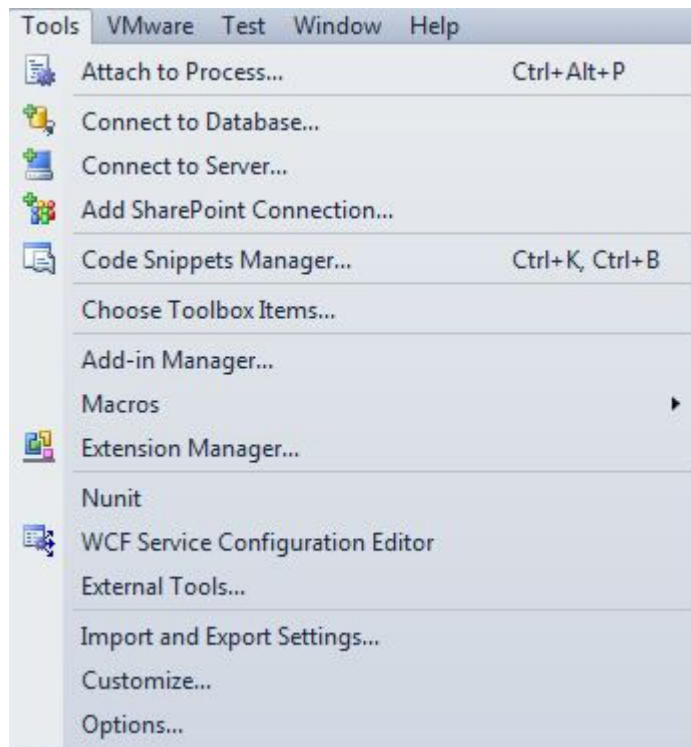


Chọn tiếp các bước cho đến khi hoàn thành

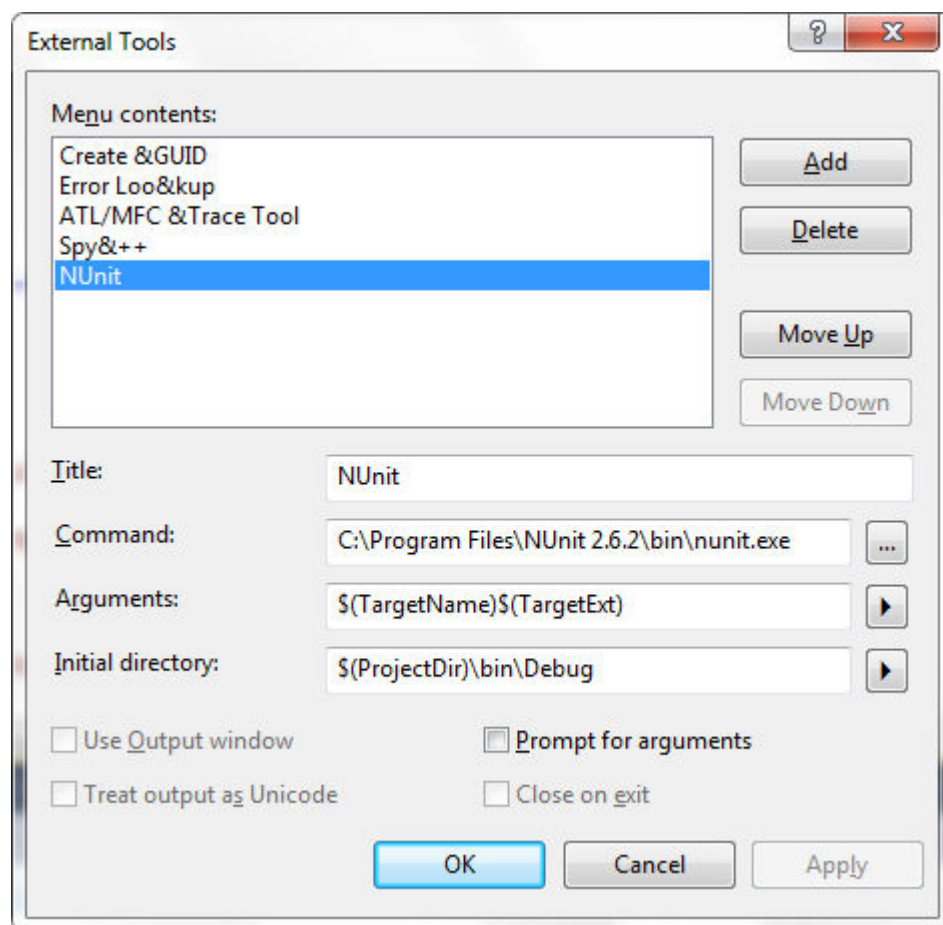
Có thể chọn Setup type là Typical hay Complete.

2.3. Add tool NUNIT vào Visual Studio

Vào Tool, chọn mục External Tools....:



Hộp thoại xuất hiện, chọn Add sau đó tiến hành điền tên của tool cần add vào mục Title và một số thông số như hình sau.

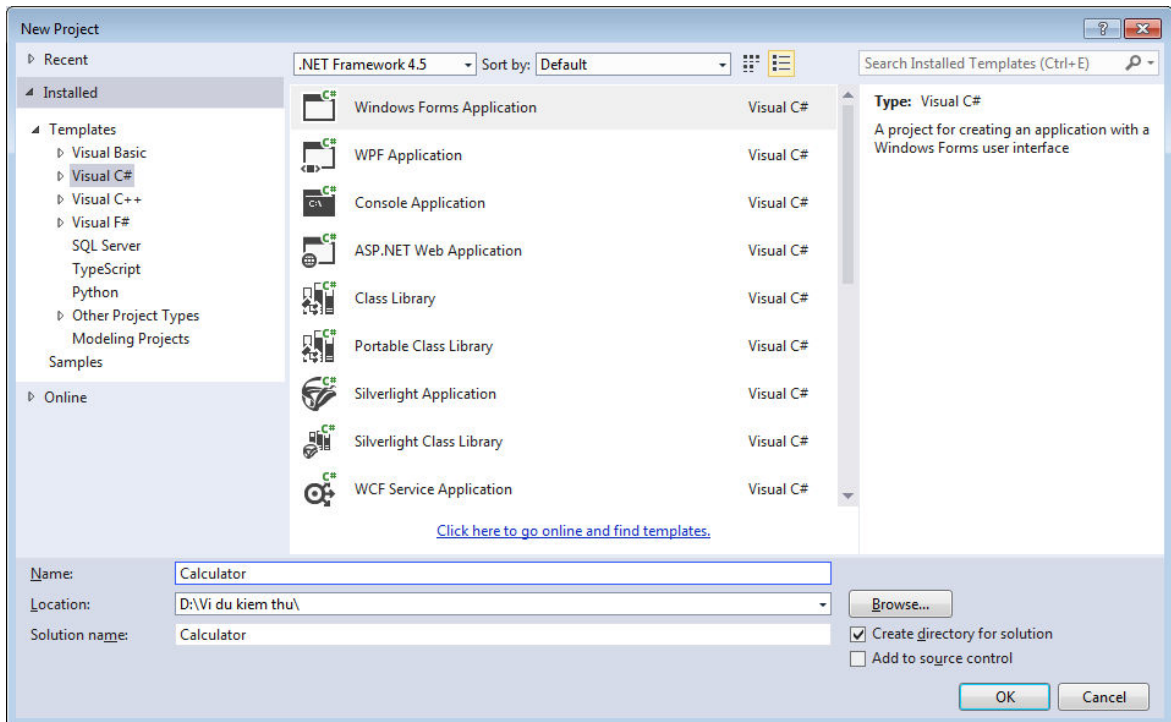


2.4. Sử dụng

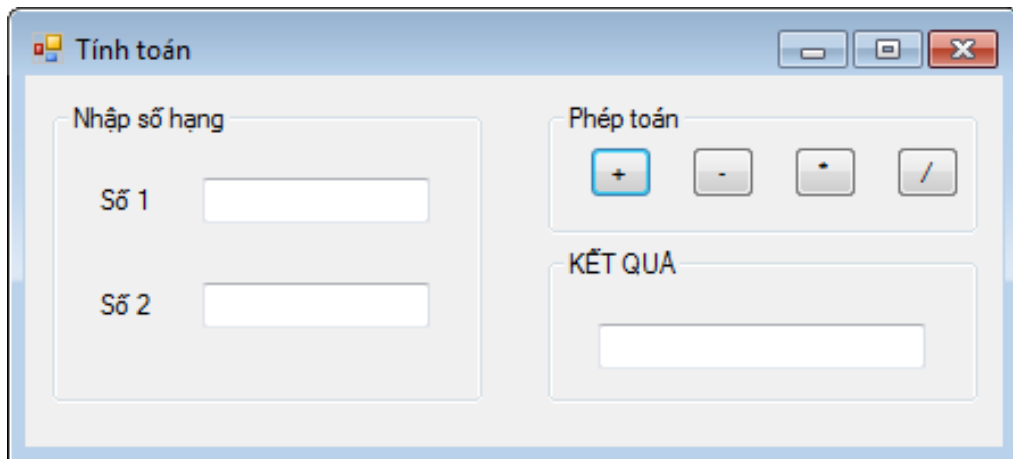
Giả sử ta được giao nhiệm vụ viết 1 lớp có tên là *Arithmetica* có 4 phương thức để thực thi các nhiệm vụ cộng, trừ, nhân và chia các số nguyên - *Add*, *Subtract*, *Multiply* và

Divide.

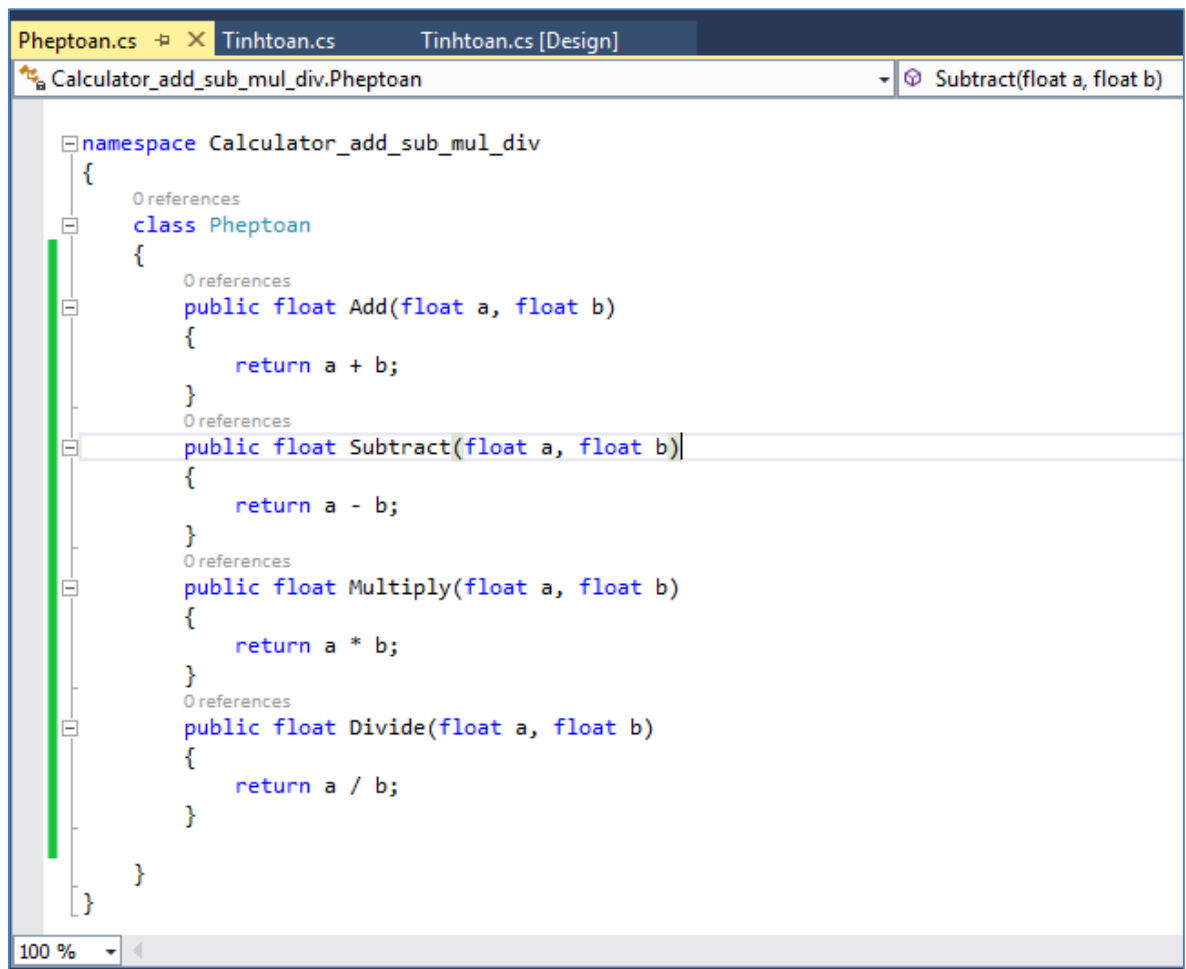
- (1) Mở Visual Studio 2013, tạo 1 project có tên là Calculator:



- (2) Tạo giao diện form như hình:



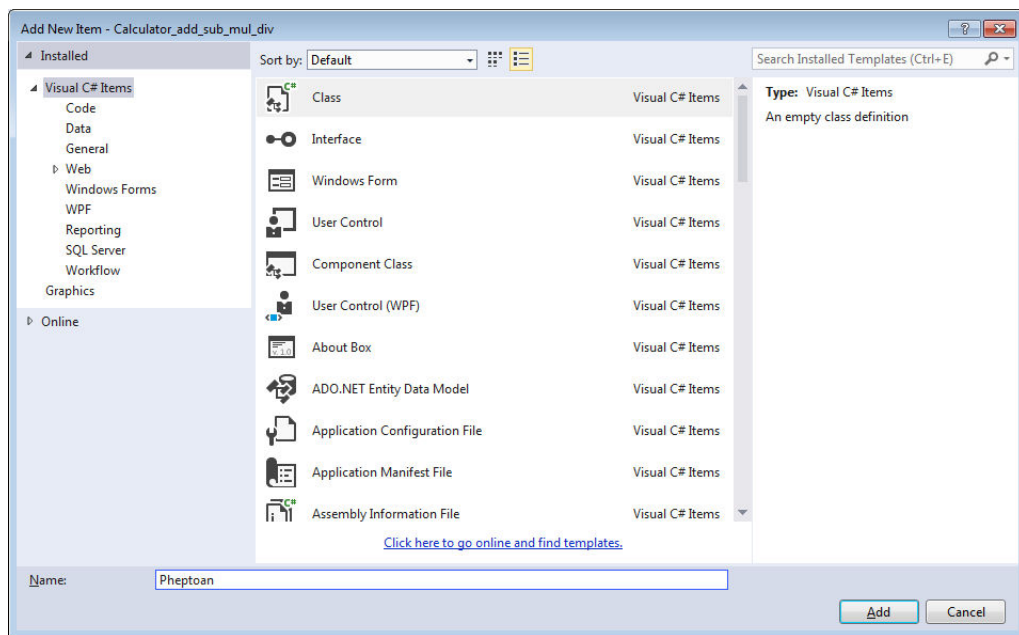
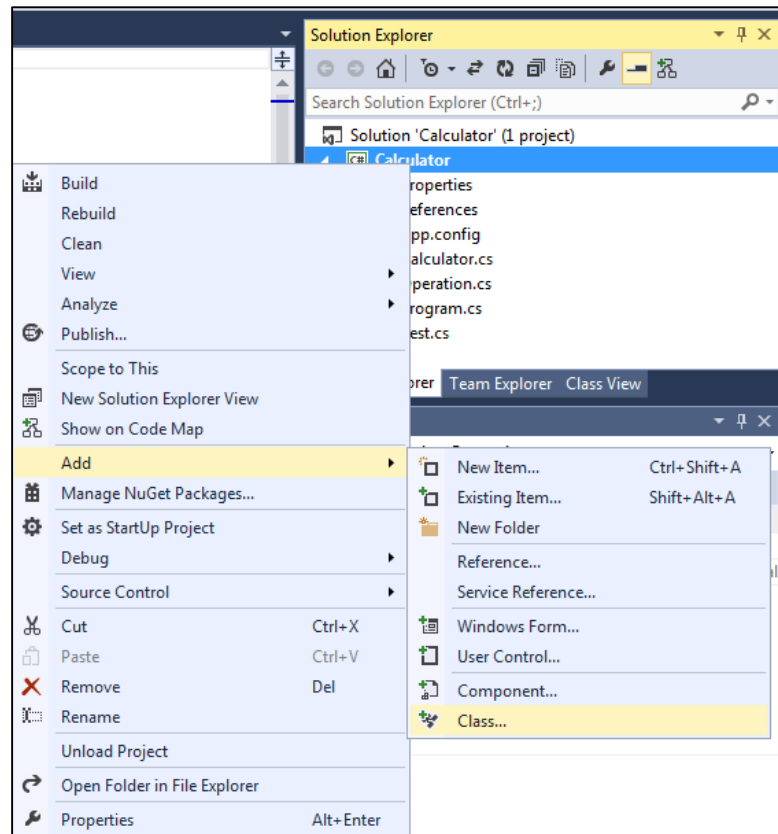
- (3) Gõ vào đoạn code như hình:



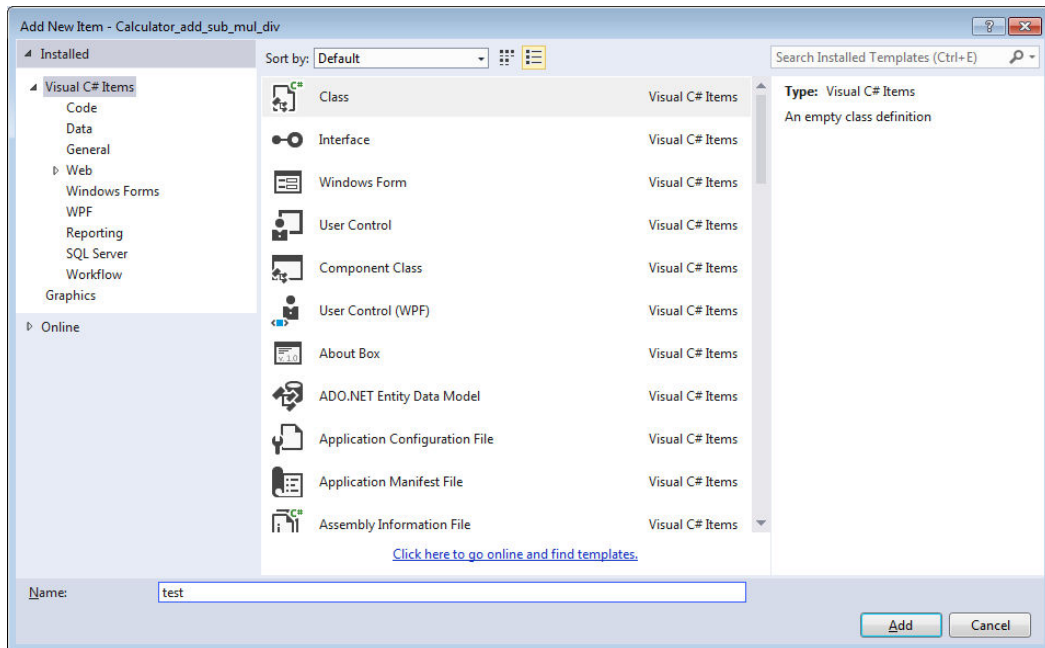
```
namespace Calculator_add_sub_mul_div
{
    0 references
    class Pheptoan
    {
        0 references
        public float Add(float a, float b)
        {
            return a + b;
        }
        0 references
        public float Subtract(float a, float b)
        {
            return a - b;
        }
        0 references
        public float Multiply(float a, float b)
        {
            return a * b;
        }
        0 references
        public float Divide(float a, float b)
        {
            return a / b;
        }
    }
}
```

Lưu ý:

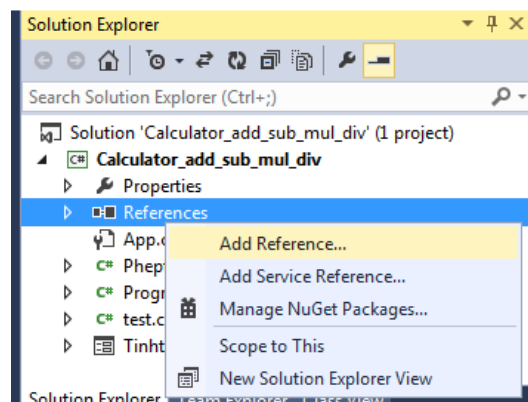
- Đoạn code này có thể gõ trực tiếp trong class .cs của form trên
- Hoặc gõ trong một class mới:
 - + Add một class mới: Phải chuột vào Project → Add → Class → Đặt tên cho class (Pheptoan)



- (4) Biên dịch và sửa lỗi cú pháp nếu có.
- (5) Sau khi viết xong class Pheptoan (hoặc viết trong class), ta tiến hành test trên các hàm vừa viết. Các thao tác như sau:
- (6) Tạo 1 class mới để test (đặt tên là: test)



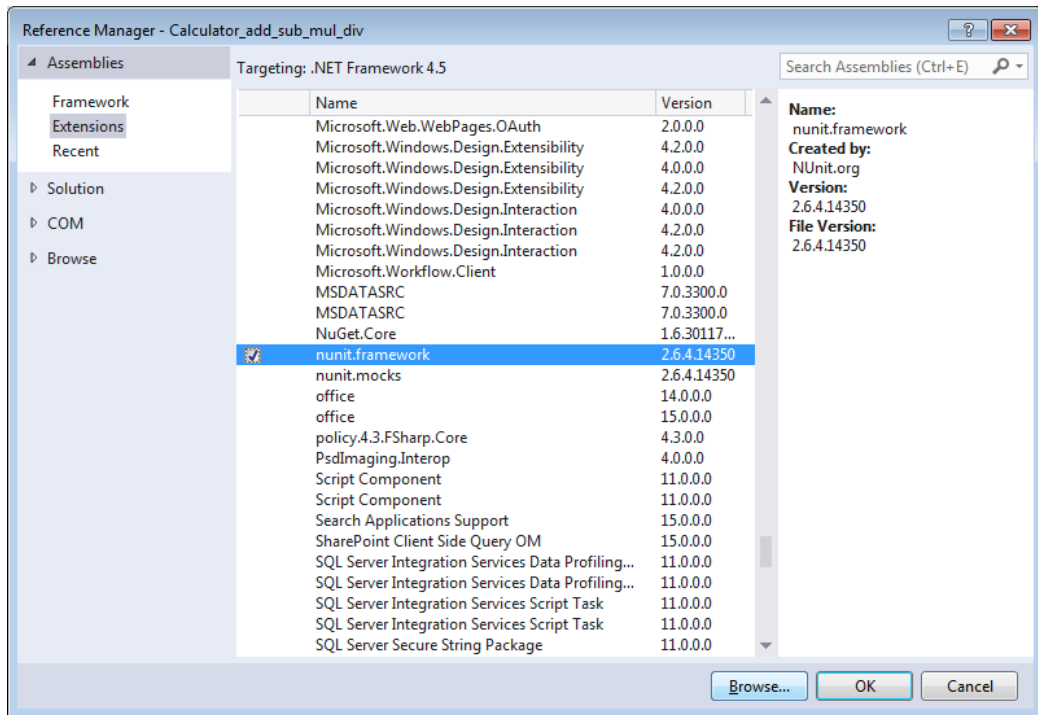
(7) Để xây dựng được các phương thức test chúng ta phải add thư viện Nunit vào: Trong cửa sổ Solution Explorer: click chuột phải vào mục References → Add reference.



Chọn mục Extension bên trái cửa sổ rồi chọn thư viện unit.framework và nhấn OK (tham khảo hình dưới).

Lưu ý:

- Nếu không tìm thấy *nunit.framework* thì nhấn thẻ *Browse* và tìm trong thư mục bin trong thư mục cài đặt của *nunit*
- Nếu sử dụng *Visual Studio 2010* trở về trước thì chọn thư viện *nunit.framework* trong tab *.NET*.



(8) Thêm dòng sau vào phần khai báo các thư viện trong class test

using NUnit.Framework;

(9) Thêm Attribute [TestFixture] vào trước class test:

[TestFixture]

public class test

{

}

[TestFixture] dùng để báo cho Nunit biết đây lớp unit test

(10) Để tiến hành test, ta phải tạo 1 đối tượng test. Công việc khởi tạo các đối tượng dùng để test sẽ nằm trong hàm Setup() có thuộc tính [TestFixtureSetUp]. Sau khi test xong, các đối tượng phải được hủy. Công việc hủy các đối tượng sau khi kiểm tra xong sẽ nằm trong hàm TearDown() với thuộc tính [TestFixtureTearDown]:

```
namespace Calculator_add_sub_mul_div
{
    [TestFixture]
    0 references
    class test
    {
        private Pheptoan pt;
        [TestFixtureSetUp]
        0 references
        public void Setup()
        {
            pt = new Pheptoan();
        }
        [TestFixtureTearDown]
        0 references
        public void TearDown()
        {
            pt = null;
        }
    }
}
```


Lưu ý: Nếu không tạo class Pheptoan thì các khai báo ở trên sẽ được thay cho class của form.

(11) Tiếp theo: Xây dựng script để test các hàm ở class pheptoan. Mọi hàm kiểm tra đều phải có thuộc tính [Test]

```
[Test]
0 references
public void test_add()
{
    Assert.AreEqual(pt.Add(2,5),7);
}
```

Trong hàm test_add phương thức **Assert.AreEqual(pt.Add(2,5),7)** được sử dụng để test. Khi phương thức này gọi sẽ gọi hàm Add với 2 tham số 2 và 5, kết quả của hàm này được kiểm tra xem có bằng 7 hay không? Nếu đúng thì hàm test này sẽ **Pass** nếu sai sẽ **Failed**

Tương tự, ta viết các hàm test_subtract, test_multiply, test_devide:

```
[Test]
0 references
public void test_subtract()
{
    Assert.AreEqual(pt.Subtract(6, 2), 4);
}

[Test]
0 references
public void test_multiply()
{
    Assert.AreEqual(pt.Multiply(3, 23), 69);
}

[Test, ExpectedException(typeof(DivideByZeroException))]
0 references
public void test_devide()
{
    Assert.AreEqual(pt.Divide(6, 3), 2);
    Assert.AreEqual(pt.Divide(2,0), 2);
}
```

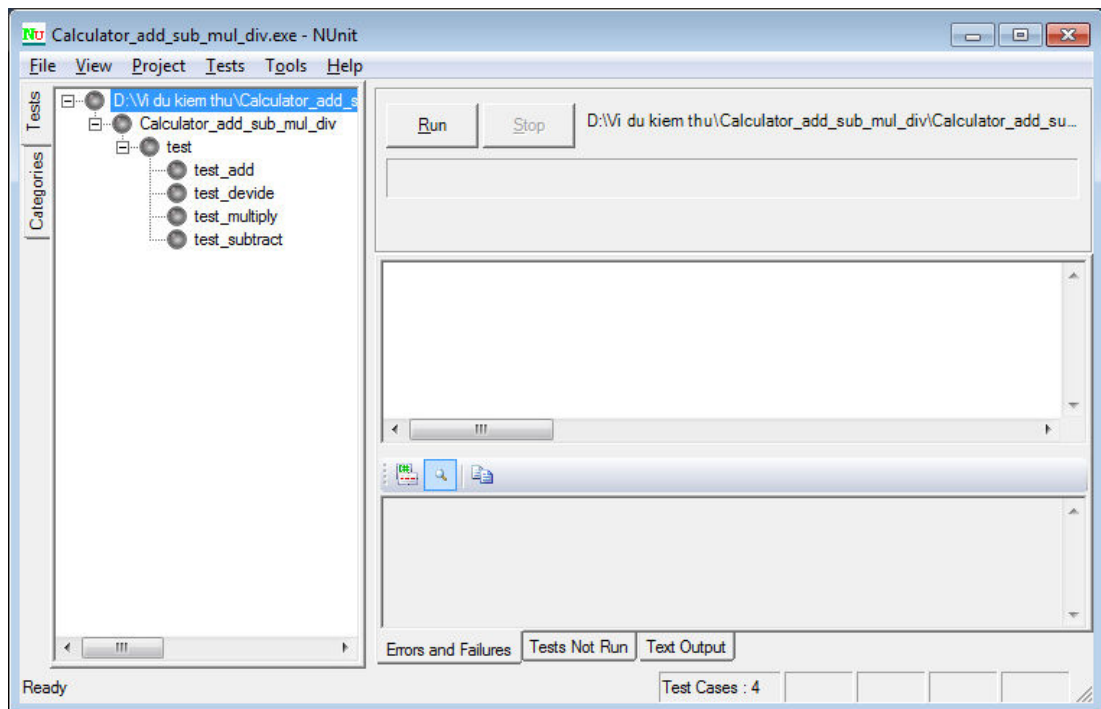
Thuộc tính: **ExpectedException(typeof(DivideByZeroException))** dùng để báo cho Nunit biết phương thức Divide có thể trả về 1 exception là DivideByZeroException. Khi exception đó xảy ra test case vẫn Pass. Lưu ý: phải sử dụng đúng tên của exception, nếu ở trên sử dụng Exception thay cho DivideByZeroException thì test case sẽ Failed.

Cú pháp của thuộc tính trên là: **[ExpectedException(<Loại exception>)]**

(12) Thực hiện biên dịch thành file dll hoặc exe.

2.5. Thực thi test

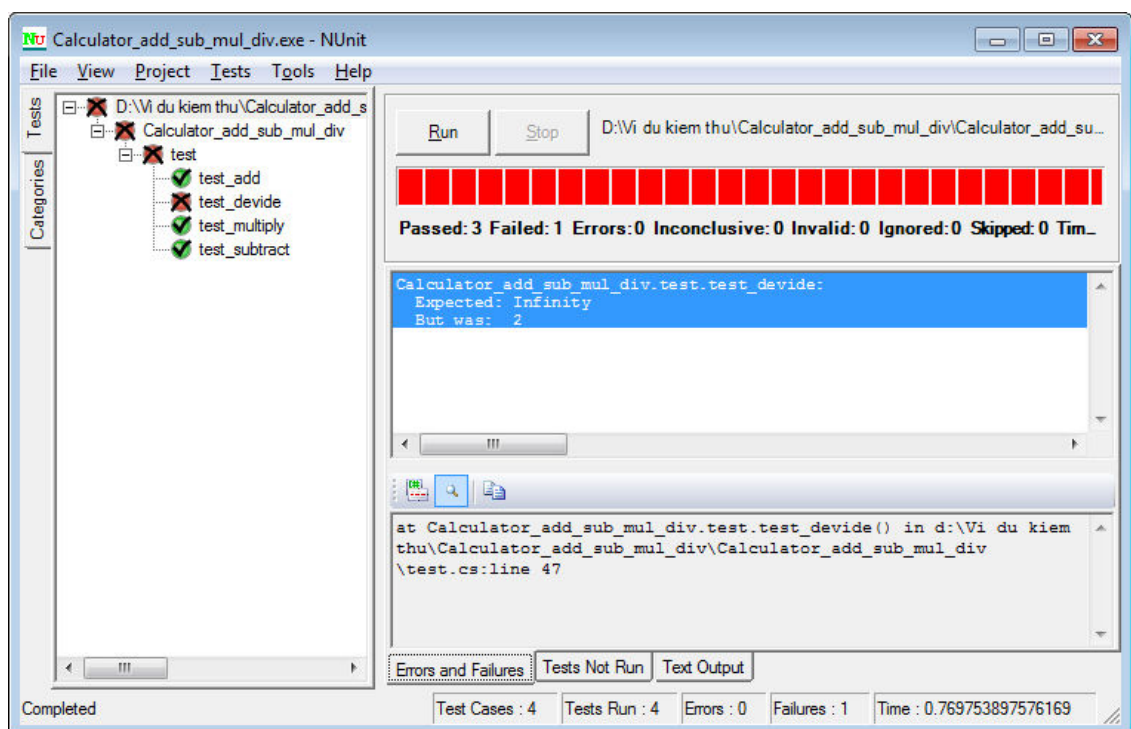
(1) Chạy nunit gui: Menu Tool (Visual Studio) → Chọn Nunit, ta có màn hình:



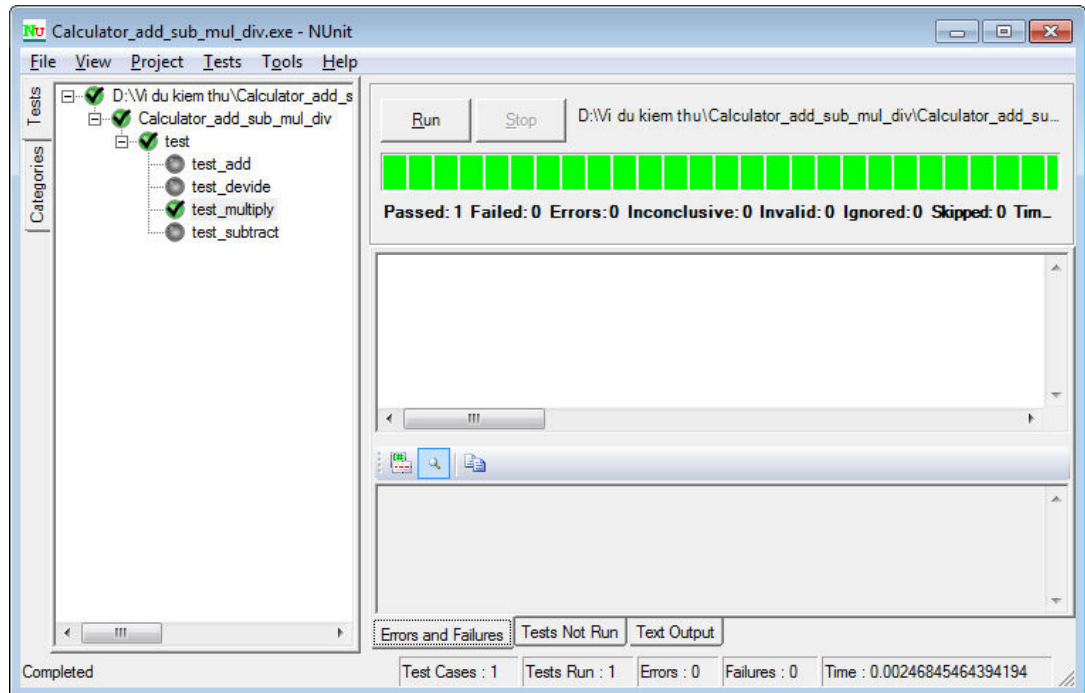
Ta sẽ thấy các hàm test bên trái cửa sổ

(2) Chạy hết tất cả các test: Nhấn nút Run

- Test nào pass thì sẽ có dấu chọn màu xanh
- Test nào fail thì sẽ có dấu nhân màu đỏ

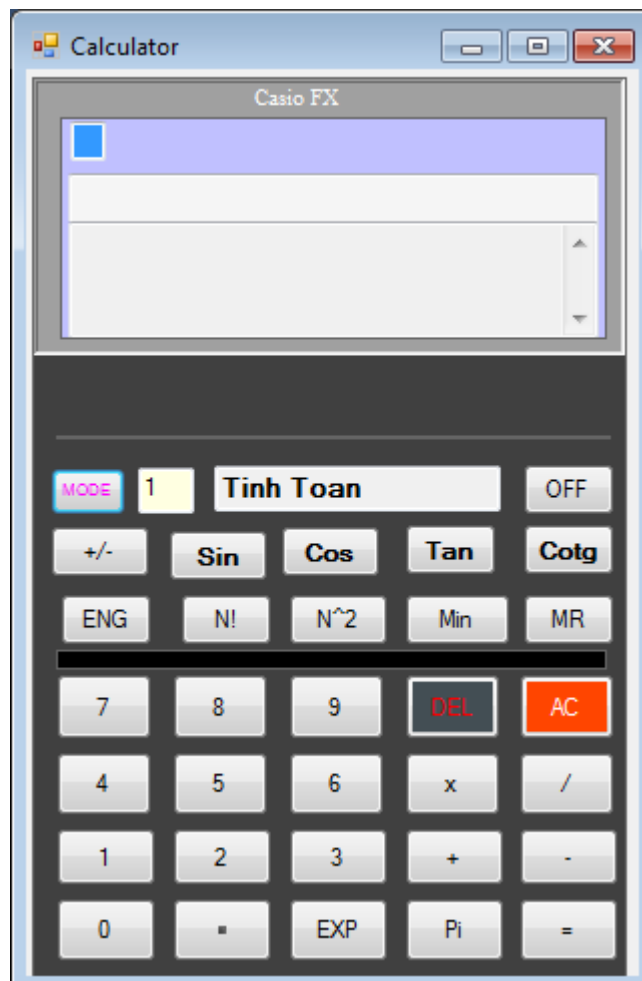


(3) Chạy từng test: Chọn test trong danh sách bên trái, nhấn nút Run



3. Yêu cầu bài tập

Từ project Calculator được cung cấp sẵn với form có giao diện như sau:



- Sinh viên tạo file excel TH1_HoVaTen_MaSoSinhVien_Unit_TestCase.xlsx các sheet trong file này đặt tên tương ứng với tên các hàm được cung cấp trong Project trên. Nội dung trình bày trong các sheet như mẫu trong file Baitapthuchanh1-Template_Unit_TestCase.xlsx
- Hãy xây dựng 1 project HoTenSinhVien_MaSoSinhVien_Test trong cùng Solution với Project trên với các class MSSV_Test sau (mỗi class thực hiện hết các test case đã mô tả trong file TH1_HoVaTen_MaSoSinhVien_Unit_TestCase.xlsx:
 - Giải phương trình bậc 2
 - Giải hệ phương trình bậc nhất hai ẩn
 - Giải hệ phương trình bậc nhất ba ẩn
 - Test hàm mũ, hàm giai thừa, hàm Sin

4. Các yêu cầu & quy định chi tiết cho bài nộp

- Làm riêng từng sinh viên
- Tập hợp các file (Project và File excel) vào chung 1 thư mục đặt tên là

TH1_HoVaTen_MaSoSinhVien. Nén thư mục này lại thành một file rar.