MỤC LỤC

[B1 :Giả sử tạo một project C# thực hiện các phép toán đơn giản cộng, trừ, nhân, chia các số Nguyên 2](#_Toc191457295)

[Đặt tên cho project và bấm next 2](#_Toc191457296)

[B2: Thiết kế giao diện tính toán như bên dưới và việc xử lý các phép tính đều thông qua lớp PhepTinh\_21\_Anh 3](#_Toc191457297)

[B3 : Tạo một tập tin PhepTinh\_21\_Anh.cs chứa lớp public PhepToan\_21\_Anh dùng có phương thức Execute để thực hiện phép tính đơn giản với hai số nguyên. 4](#_Toc191457298)

[Tạo project kiểm thử để kiểm thử các phép toán trong chương trình trên. Click chuột phải Solution > Add > New Projects... 4](#_Toc191457299)

[Tạo mới 1 class PhepToan\_21\_Anh.cs 4](#_Toc191457300)

[Code trong class PhepTinh\_21\_Anh 5](#_Toc191457301)

[Vào winform 6](#_Toc191457302)

[B4:Tạo project kiểm thử để kiểm thử các phép toán trong chương trình trên. Click chuột phải Solution > Add > New Projects... Sau đó chọn loại project là “Unit Test Project” và đặt tên là PhepToanTester\_21\_Anh 8](#_Toc191457303)

[Đặt tên cho Project UnitTest 9](#_Toc191457304)

[Tại project Unit Test, thực hiện Add Reference để tham chiếu đến project cần thực hiện Unit Test 10](#_Toc191457305)

[Chọn project PhepTinh\_21\_Anh để test. 10](#_Toc191457306)

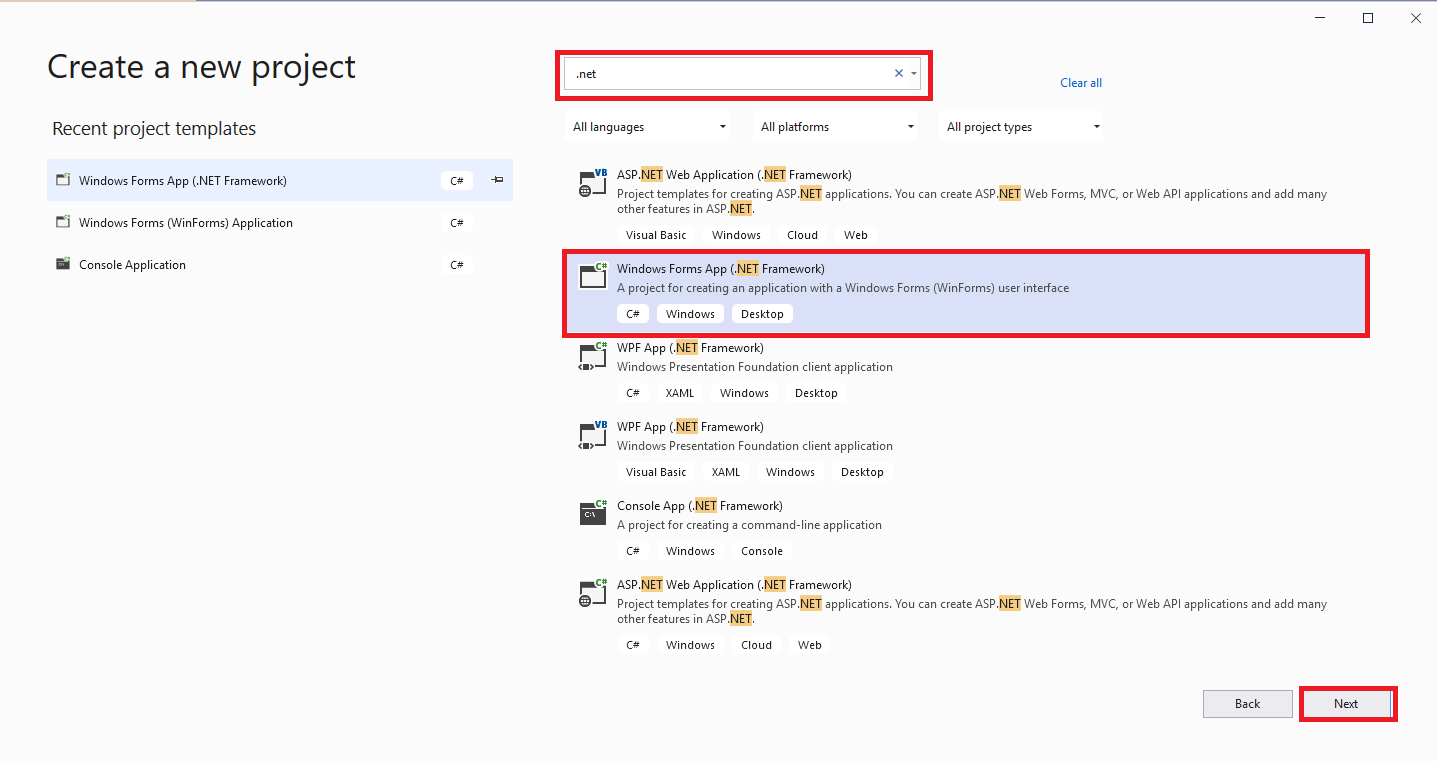
[B5:Viết code kiểm thử phương thức Execute trong lớp PhepTinh\_21\_Anh 11](#_Toc191457307)

[B6:Chạy các unit test đã viết: 16](#_Toc191457308)

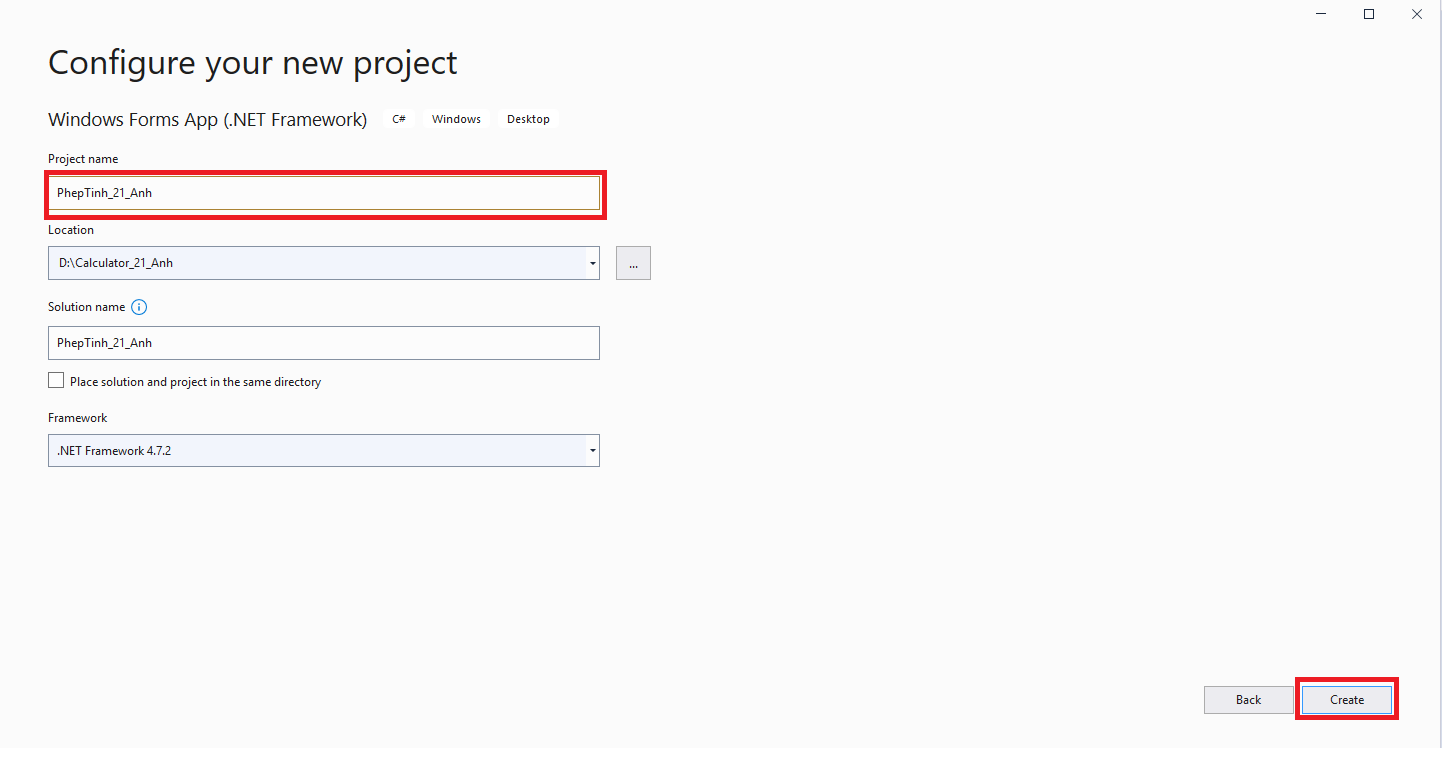
[Bấm vào đây để chạy tất cả testcase hiện có 17](#_Toc191457309)

[B7: Kết quả các testcase 17](#_Toc191457310)

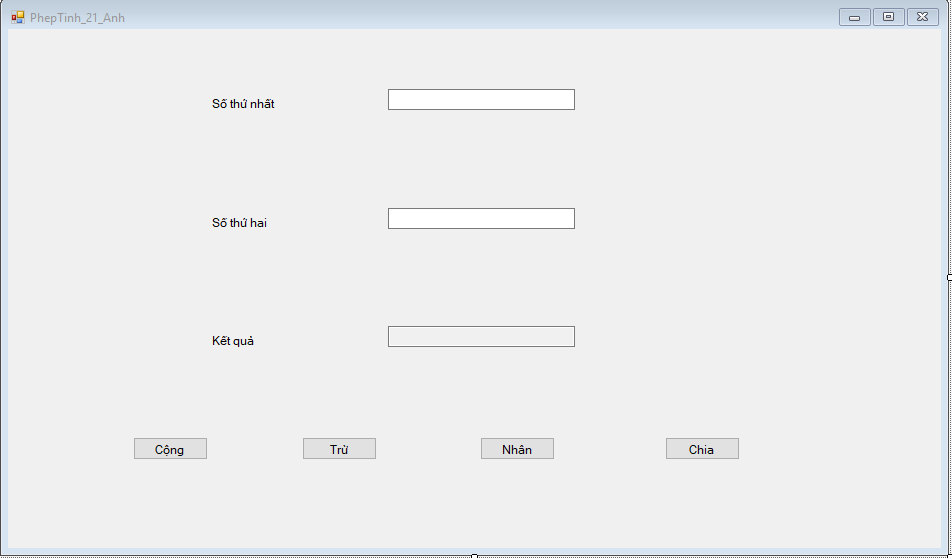
# B1 :Giả sử tạo một project C# thực hiện các phép toán đơn giản cộng, trừ, nhân, chia các số Nguyên



## Đặt tên cho project và bấm next



# B2: Thiết kế giao diện tính toán như bên dưới và việc xử lý các phép tính đều thông qua lớp PhepTinh\_21\_Anh

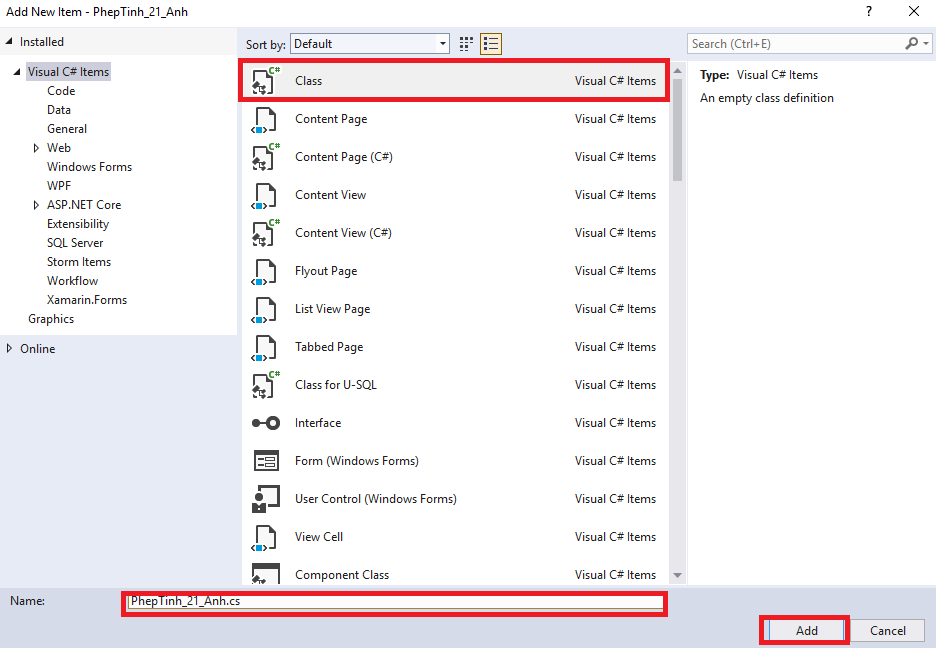


# B3 : Tạo một tập tin PhepTinh\_21\_Anh.cs chứa lớp public PhepToan\_21\_Anh dùng có phương thức Execute để thực hiện phép tính đơn giản với hai số nguyên.

## Tạo project kiểm thử để kiểm thử các phép toán trong chương trình trên. Click chuột phải Solution > Add > New Projects...



## Tạo mới 1 class PhepToan\_21\_Anh.cs



## Code trong class PhepTinh\_21\_Anh

namespace PhepTinh\_21\_Anh

{

//Lưu ý phải để public vì mặc định access modifier của c# là internal -> class không truyền khác project được

public class PhepToan\_21\_Anh

{

private int a\_21\_Anh, b\_21\_Anh;

public PhepToan\_21\_Anh(int a\_21\_Anh, int b\_21\_Anh)

{

this.a\_21\_Anh = a\_21\_Anh;

this.b\_21\_Anh = b\_21\_Anh;

}

public int Exucute\_21\_Anh(string calSymbol\_21\_Anh)

{

int res\_21\_Anh = 0;

switch (calSymbol\_21\_Anh)

{

case "+":

res\_21\_Anh = a\_21\_Anh + b\_21\_Anh;

break;

case "-":

res\_21\_Anh = a\_21\_Anh - b\_21\_Anh;

break;

case "\*":

res\_21\_Anh = a\_21\_Anh \* b\_21\_Anh;

break;

case "/":

res\_21\_Anh = a\_21\_Anh / b\_21\_Anh;

break;

}

return res\_21\_Anh;

}

}

}

## Vào winform

private void btnCong\_21\_Anh\_Click(object sender, EventArgs e)

{

int a\_21\_Anh = int.Parse(txtSo1\_21\_Anh.Text);

int b\_21\_Anh = int.Parse(txtSo2\_21\_Anh.Text);

PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(a\_21\_Anh, b\_21\_Anh);

int ketQua\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("+");

txtKetQua\_21\_Anh.Text = ketQua\_21\_Anh.ToString();

}

private void btnTru\_21\_Anh\_Click(object sender, EventArgs e)

{

int a\_21\_Anh = int.Parse(txtSo1\_21\_Anh.Text);

int b\_21\_Anh = int.Parse(txtSo2\_21\_Anh.Text);

PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(a\_21\_Anh, b\_21\_Anh);

int ketQua\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("-");

txtKetQua\_21\_Anh.Text = ketQua\_21\_Anh.ToString();

}

private void btnNhan\_21\_Anh\_Click(object sender, EventArgs e)

{

int a\_21\_Anh = int.Parse(txtSo1\_21\_Anh.Text);

int b\_21\_Anh = int.Parse(txtSo2\_21\_Anh.Text);

PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(a\_21\_Anh, b\_21\_Anh);

int ketQua\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("\*");

txtKetQua\_21\_Anh.Text = ketQua\_21\_Anh.ToString();

}

private void btnChia\_21\_Anh\_Click(object sender, EventArgs e)

{

int a\_21\_Anh = int.Parse(txtSo1\_21\_Anh.Text);

int b\_21\_Anh = int.Parse(txtSo2\_21\_Anh.Text);

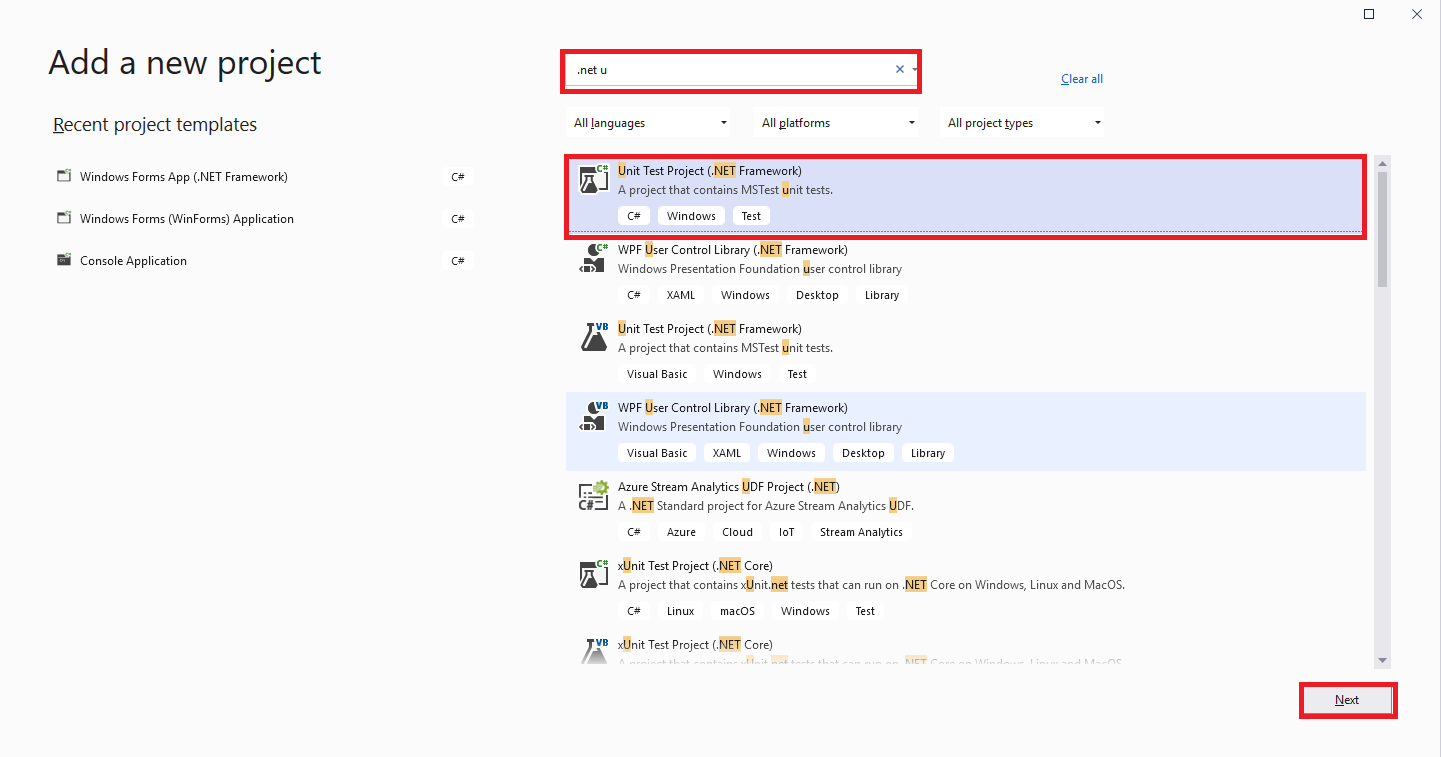
PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(a\_21\_Anh, b\_21\_Anh);

int ketQua\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("/");

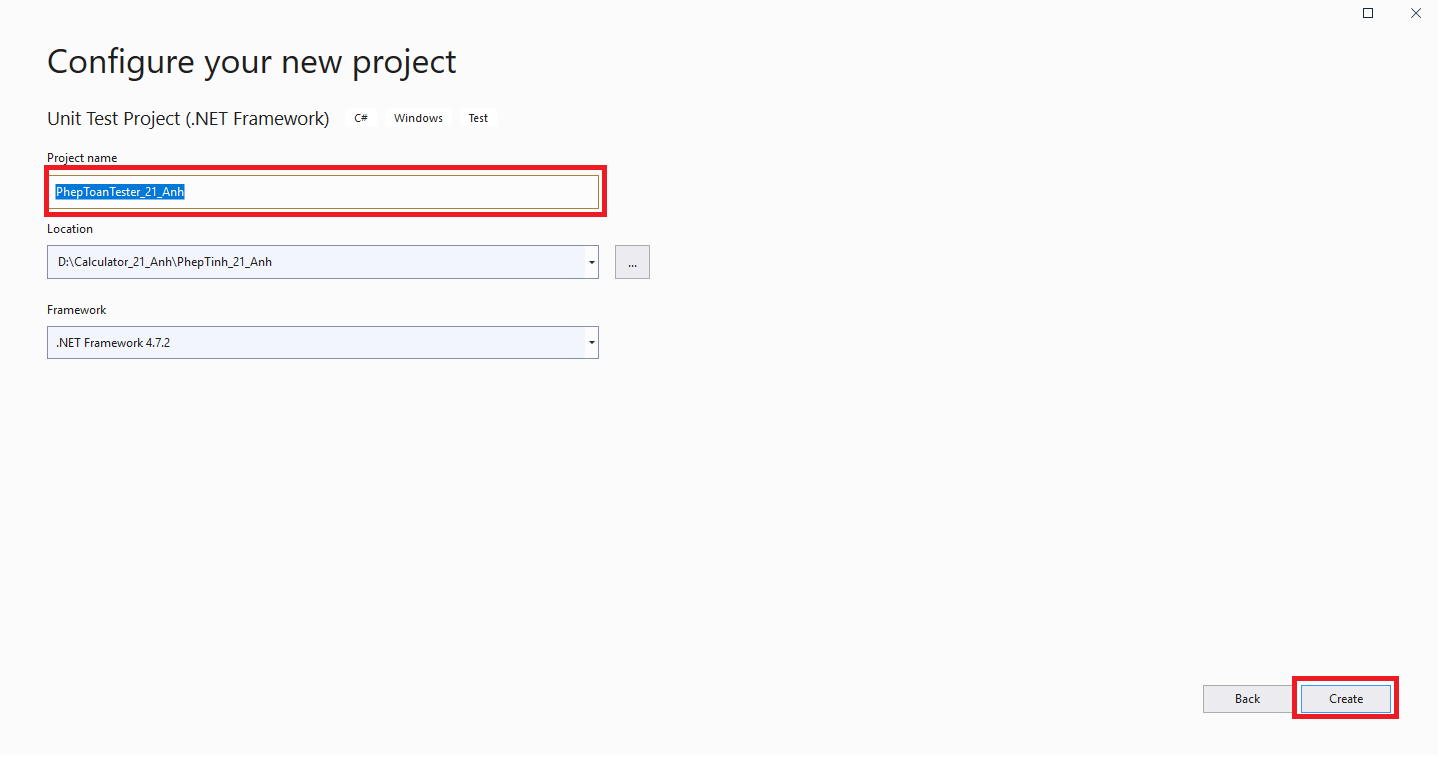
txtKetQua\_21\_Anh.Text = ketQua\_21\_Anh.ToString();

}

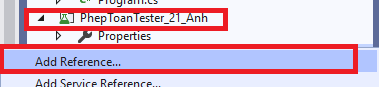
# B4:Tạo project kiểm thử để kiểm thử các phép toán trong chương trình trên. Click chuột phải Solution > Add > New Projects... Sau đó chọn loại project là “Unit Test Project” và đặt tên là PhepToanTester\_21\_Anh



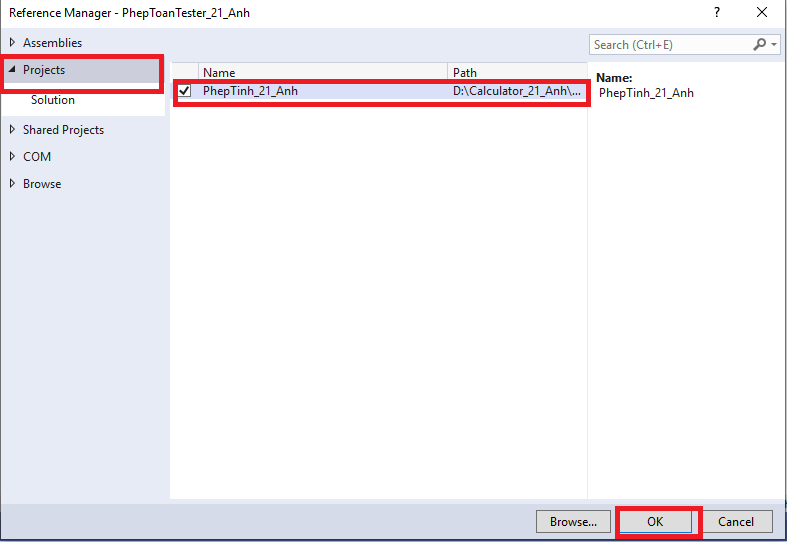
## Đặt tên cho Project UnitTest



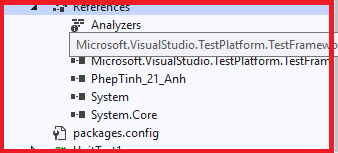
## Tại project Unit Test, thực hiện Add Reference để tham chiếu đến project cần thực hiện Unit Test



## Chọn project PhepTinh\_21\_Anh để test.



# B5:Viết code kiểm thử phương thức Execute trong lớp PhepTinh\_21\_Anh



namespace PhepToanTester\_21\_Anh

{

[TestClass]

public class UnitTest\_PhepToan\_21\_Anh

{

private PhepToan\_21\_Anh pt\_21\_Anh;

[TestInitialize]

public void Init\_21\_Anh()//Hàm dùng chung Cho các TestCase

{

pt\_21\_Anh = new PhepToan\_21\_Anh(8, 2);

}

[TestMethod]

public void TC1\_Cong\_fail\_21\_Anh() //TC1\_Cong\_fail\_21\_Anh a= 8 b=2 ketqua=1

{

int excepted\_21\_Anh = 1;

//PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(8, 2);

int actual\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("+");

Assert.AreEqual(excepted\_21\_Anh, actual\_21\_Anh);

}

[TestMethod]

public void TC2\_Tru\_fail\_21\_Anh() //TC2\_Tru\_fail\_21\_Anh a= 8 b=2 ketqua=8

{

int excepted\_21\_Anh = 8;

//PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(8, 2);

int actual\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("-");

Assert.AreEqual(excepted\_21\_Anh, actual\_21\_Anh);

}

[TestMethod]

public void TC3\_Nhan\_fail\_21\_Anh() //TC3\_Nhan\_fail\_21\_Anh a=8 b=2 ketqua =6

{

int excepted\_21\_Anh = 6;

//PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(8, 2);

int actual\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("\*");

Assert.AreEqual(excepted\_21\_Anh, actual\_21\_Anh);

}

[TestMethod]

public void TC4\_Chia\_fail\_21\_Anh() //TC4\_Chia\_fail\_21\_Anh a=8 b=2 ketqua =48

{

int excepted\_21\_Anh = 48;

//PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(8, 2);

int actual\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("/");

Assert.AreEqual(excepted\_21\_Anh, actual\_21\_Anh);

}

[ExpectedException(typeof(DivideByZeroException))]

[TestMethod]

public void TC5\_ChiaChoKhong\_fail\_21\_Anh() //TC5\_ChiaChoKhong\_fail\_21\_Anh -> kết quả in ra ngoại lệ

{

PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(8, 2);

pt\_21\_Anh.Exucute\_21\_Anh("/");

}

[TestMethod]

public void TC6\_Cong\_21\_Anh() //TC6\_Cong\_21\_Anh a= 8 b=2 ketqua=10

{

int excepted\_21\_Anh = 10;

//PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(8, 2);

int actual\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("+");

Assert.AreEqual(excepted\_21\_Anh, actual\_21\_Anh);

}

[TestMethod]

public void TC7\_Tru\_21\_Anh() //TC7\_Tru\_21\_Anh a= 8 b=2 ketqua=6

{

int excepted\_21\_Anh = 6;

//PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(8, 2);

int actual\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("-");

Assert.AreEqual(excepted\_21\_Anh, actual\_21\_Anh);

}

[TestMethod]

public void TC8\_Nhan\_21\_Anh() //TC8\_Nhan\_21\_Anh a=8 b=2 ketqua =16

{

int excepted\_21\_Anh = 16;

//PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(8, 2);

int actual\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("\*");

Assert.AreEqual(excepted\_21\_Anh, actual\_21\_Anh);

}

[TestMethod]

public void TC9\_Chia\_21\_Anh() //TC9\_Chia\_21\_Anh a=8 b=2 ketqua =4

{

int excepted\_21\_Anh = 4;

//PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(8, 2);

int actual\_21\_Anh = pt\_21\_Anh.Exucute\_21\_Anh("/");

Assert.AreEqual(excepted\_21\_Anh, actual\_21\_Anh);

}

[ExpectedException(typeof(DivideByZeroException))]

[TestMethod]

public void TC10\_ChiaChoKhong\_21\_Anh() //TC10\_ChiaChoKhong\_21\_Anh ->kết quả in ra ngoại lệ

{

PhepToan\_21\_Anh pt\_21\_Anh = new PhepToan\_21\_Anh(8, 0);

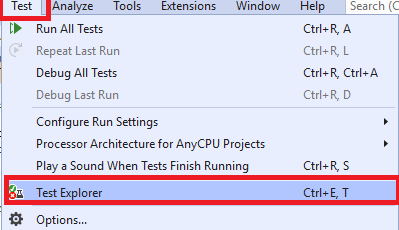
pt\_21\_Anh.Exucute\_21\_Anh("/");

}

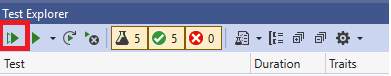
}

}

# B6:Chạy các unit test đã viết:



## Bấm vào đây để chạy tất cả testcase hiện có



# B7: Kết quả các testcase

