Nội dung

[**1. Tạo Project C# với các phép toán đơn giản** 1](#_Toc190763171)

[Bước 1: Tạo một dự án C# WinForms 1](#_Toc190763172)

[Bước 2: Thiết kế giao diện người dùng 1](#_Toc190763173)

[Bước 3: Tạo lớp Calculation 1](#_Toc190763174)

[Bước 4: Xử lý sự kiện nút tính toán 3](#_Toc190763175)

[**2. Tạo Project Kiểm thử (Unit Test Project)** 6](#_Toc190763176)

[Bước 1: Thêm dự án kiểm thử 6](#_Toc190763177)

[Bước 2: Thêm tham chiếu đến dự án chính 6](#_Toc190763178)

[Bước 3: Viết mã kiểm thử với MSTest 6](#_Toc190763179)

[Bước 4: Chạy Unit Tests 7](#_Toc190763180)

Các bước làm

Dưới đây là hướng dẫn từng bước để thực hiện dự án tính toán đơn giản và kiểm thử với MS Unit và NUNIT trong C#:

# **1. Tạo Project C# với các phép toán đơn giản**

## **Bước 1: Tạo một dự án C# WinForms**

1. Mở Visual Studio, tạo một dự án mới chọn Windows Forms App (.NET).
2. Đặt tên dự án là Calculator.

## **Bước 2: Thiết kế giao diện người dùng**

A screenshot of a computer

AI-generated content may be incorrect.

1. Thêm các điều khiển vào form:
   * Hai TextBox để nhập các số nguyên (lb1\_hau\_4, lb2\_hau\_4).
   * Một TextBox để hiển thị kết quả (lb3\_hau\_4).
   * Một số nút bấm (Button) cho các phép toán như cộng, trừ, nhân, chia (cong\_hau\_4, tru\_hau\_4, nhan\_hau\_4, chia\_hau\_4).

Giao diện sẽ có các phần như sau:

* + Nhập hai số nguyên.
  + Hiển thị kết quả.
  + Các nút tính toán tương ứng với các phép toán.

## **Bước 3: Tạo lớp Calculation**

1. Thêm một lớp mới Calculation\_hau\_4.cs vào dự án.
2. Định nghĩa lớp Calculation với các phương thức thực hiện phép tính:

public class Calculation

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Calculator

{

public class Caculation

{

private int a, b;

public Caculation(int a, int b)

{

this.a = a;

this.b = b;

}

public int Execute(string CalSymbol)

{

int result = 0;

switch (CalSymbol)

{

case "+":

result = this.a + this.b;

break;

case "-":

result = this.a - this.b;

break;

case "\*":

result = this.a \* this.b;

break;

case "/":

result = this.a / this.b;

break;

}

return result;

}

}

}

## **Bước 4: Xử lý sự kiện nút tính toán**

1. Thêm mã vào các sự kiện Click của nút tính toán để thực hiện phép toán thông qua lớp Calculation.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Calculator

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void cong\_hau\_4\_Click(object sender, EventArgs e)

{

int a, b, ketqua;

a = int.Parse(lb1\_hau\_4.Text);

b = int.Parse(lb2\_hau\_4.Text);

Caculation c = new Caculation(a, b);

ketqua = c.Execute("+");

lb3\_hau\_4.Text = ketqua.ToString();

}

private void tru\_hau\_4\_Click(object sender, EventArgs e)

{

int a, b, ketqua;

a = int.Parse(lb1\_hau\_4.Text);

b = int.Parse(lb2\_hau\_4.Text);

Caculation c = new Caculation(a, b);

ketqua = c.Execute("-");

lb3\_hau\_4.Text = ketqua.ToString();

}

private void nhan\_hau\_4\_Click(object sender, EventArgs e)

{

int a, b, ketqua;

a = int.Parse(lb1\_hau\_4.Text);

b = int.Parse(lb2\_hau\_4.Text);

Caculation c = new Caculation(a, b);

ketqua = c.Execute("\*");

lb3\_hau\_4.Text = ketqua.ToString();

}

private void chia\_hau\_4\_Click(object sender, EventArgs e)

{

int a, b, ketqua;

a = int.Parse(lb1\_hau\_4.Text);

b = int.Parse(lb2\_hau\_4.Text);

Caculation c = new Caculation(a, b);

ketqua = c.Execute("/");

lb3\_hau\_4.Text = ketqua.ToString();

}

}

}

Tương tự, bạn sẽ cần thêm các sự kiện Click cho các phép toán khác như trừ, nhân, chia.

# **2. Tạo Project Kiểm thử (Unit Test Project)**

## **Bước 1: Thêm dự án kiểm thử**

1. Click chuột phải vào **Solution** trong **Solution Explorer**.
2. Chọn **Add** > **New Project...** và chọn **Unit Test Project**.
3. Đặt tên cho dự án kiểm thử là CalculatorTester.

## **Bước 2: Thêm tham chiếu đến dự án chính**

1. Chuột phải vào **References** trong dự án CalculatorTester, chọn **Add Reference**.
2. Chọn dự án Calculator để tham chiếu đến lớp Calculation\_hau\_4.

## **Bước 3: Viết mã kiểm thử với MSTest**

1. Trong dự án kiểm thử, tạo một lớp kiểm thử mới. Thêm mã sau để kiểm thử phương thức Execute trong lớp Calculation\_hau\_4:
2. using Microsoft.VisualStudio.TestTools.UnitTesting;
3. using System;
4. using Calculator;
5. namespace CalculatorTester
6. {
7. [TestClass]
8. public class UnitTest1
9. {
10. private Caculation c;
11. [TestInitialize]
12. public void SetUp()
13. {
14. c = new Caculation(10, 5);
15. }
16. [TestMethod] //TC1: a =10, b = 5, kq= 15
17. public void Test\_Cong\_hau\_4()
18. {
19. int expected, actual;
20. expected = 15;
21. actual = c.Execute("+");
22. Assert.AreEqual(expected, actual);
23. }
24. [TestMethod] //TC1: a =10, b = 5, kq= 5
25. public void Test\_Tru\_hau\_4()
26. {
27. int expected, actual;
28. expected = 5;
29. actual = c.Execute("-");
30. Assert.AreEqual(expected, actual);
31. }
32. [TestMethod] //TC1: a =10, b = 5, kq= 50
33. public void Test\_Nhan\_hau\_4()
34. {
35. int expected, actual;
36. expected = 50;
37. actual = c.Execute("\*");
38. Assert.AreEqual(expected, actual);
39. }
40. [TestMethod] //TC1: a =10, b = 5, kq= 2
41. public void Test\_Chia\_hau\_4()
42. {
43. int expected, actual;
44. expected = 2;
45. actual = c.Execute("/");
46. Assert.AreEqual(expected, actual);
47. }
48. [TestMethod]
49. [ExpectedException(typeof(DivideByZeroException))]
50. public void Test\_ChiaZero\_hau\_4()
51. {
52. c = new Caculation(10, 0);
53. c.Execute("/");
54. }
55. }
56. }

## **Bước 4: Chạy Unit Tests**

1. Chạy các kiểm thử đơn vị (Unit Tests) từ **Test Explorer** trong Visual Studio.
2. Kiểm tra kết quả và xác minh rằng tất cả các phép toán đều thực hiện đúng.