



TRƯỜNG ĐẠI HỌC MỞ
TP HỒ CHÍ MINH

BÀI GIẢNG
LẬP TRÌNH GIAO DIỆN

**CHƯƠNG 1:
TỔNG QUAN VỀ LẬP TRÌNH GIAO DIỆN
VỚI VISUAL STUDIO.NET**

ThS. Nguyễn Thị Phương Trang
trang.ntp@ou.edu.vn

- Qua chương này, sinh viên có khả năng:
 - Phân biệt được các tầng và chức năng mỗi tầng trong quá trình thiết kế xây dựng phần mềm, đặc biệt là tầng giao diện.
 - Phân biệt được các loại giao diện ứng dụng để có thể lựa chọn loại giao diện phù hợp với yêu cầu sử dụng của phần mềm.
 - Biết và sử dụng được công nghệ VisualStudio.Net của Microsoft và C#
 - Làm quen và xây dựng được một ứng dụng đơn giản trong môi trường Visual Studio.Net



Nội dung chương

- Tổng quan về lập trình giao diện.
- Giới thiệu công nghệ .NET của Microsoft.





Ví dụ ứng dụng winform

TÍNH TOÁN

THỰC HIỆN CÁC PHÉP TÍNH

Nhập số n: 5

Nhập số m: 10

Kết quả: 15

+ - * / Xóa Thoát

frmNhanVien

DANH MỤC NHÂN VIÊN

Thông tin chi tiết

Họ tên	NGUYỄN MINH THÀNH	Điện thoại	0908373612
Ngày sinh	05 Tháng Tư 1983	Địa chỉ	41/4 CALMETTE Q1 TPHCM

Thêm Xóa Sửa Thoát

Thông tin chung

Họ Tên	Ngày sinh	Điện thoại	Địa chỉ
PHẠM MINH VŨ	24/01/1980 12:00:00	090564...	163/30 Thành Thái ...
NGUYỄN MINH THÀNH	05/04/1983 12:00:00	090837...	41/4 CALMETTE Q...
NGUYỄN HÀ MY	13/04/1985 12:00:00	090878...	178 NAM KÝ KHỜI ...





Khái niệm về mô hình đa tầng, đa lớp

Khái niệm về tầng

- Là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI - user interface), các quy tắc xử lý (BR - Business Rule hay BL - Business Logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng được coi là một kiến trúc phần mềm và là một mẫu thiết kế



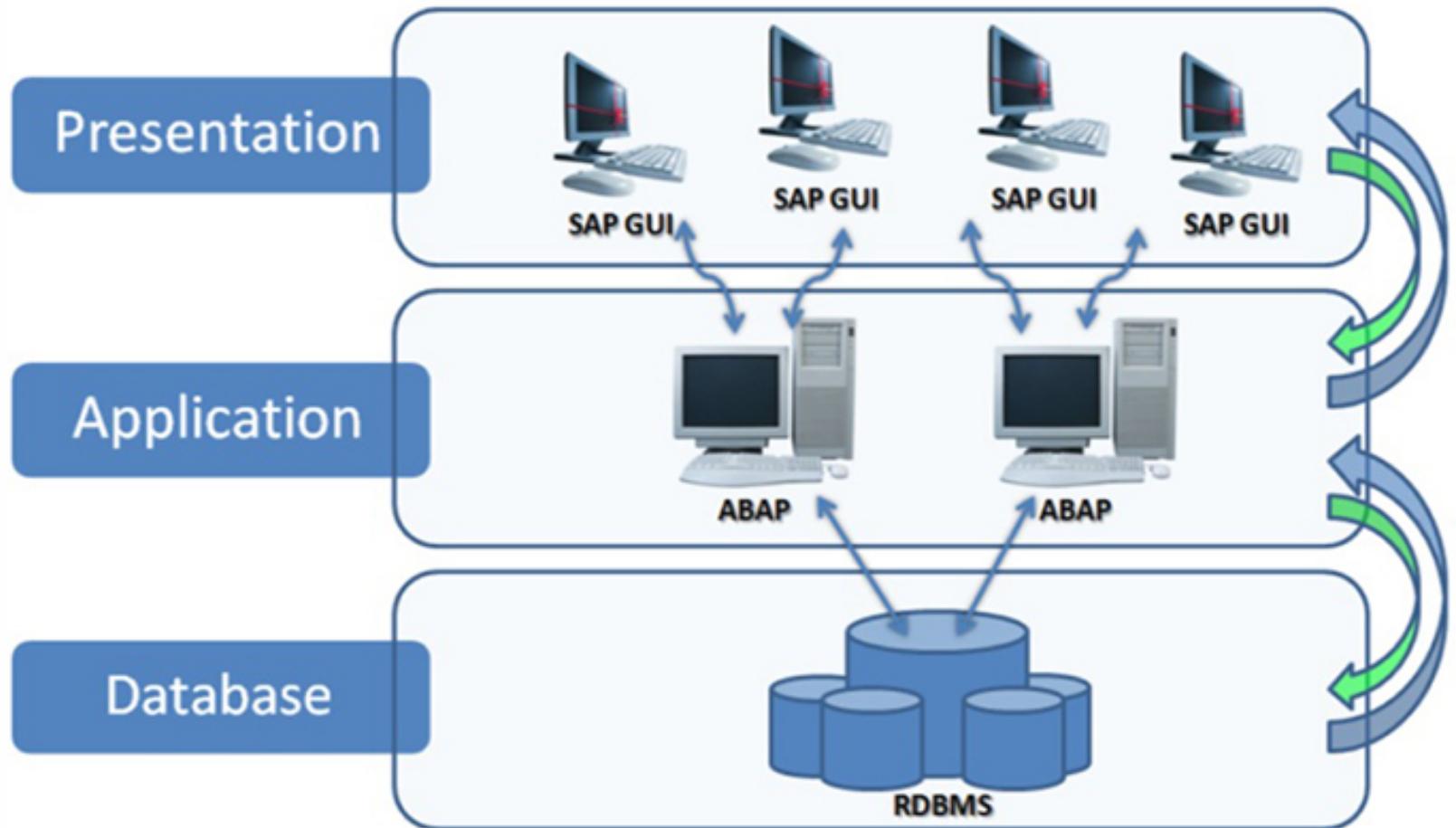
Khái niệm về mô hình đa tầng, đa lớp

- Mô hình ba tầng bao gồm:
 - **Presentation tier:** gồm các thành phần xử lý giao diện - Graphic User Interface (GUI)
 - **Business tier:** gồm các thành phần xử lý nghiệp vụ như
 - Business Logic Layer (BLL)
 - Data Access Layer (DAL)
 - Data Transfer Object (DTO)
 - **Data tier:** nơi lưu trữ dữ liệu, là các hệ quản trị CSDL như MS SQL Server, Oracle, SQLite, MS Access, XML files, text files,...



Khái niệm về mô hình đa tầng, đa lớp

Ví dụ về tầng





Khái niệm về mô hình đa tầng, đa lớp

- Mô hình đa tầng phân **chia ứng dụng** theo **mức vật lý**
- Ưu điểm của mô hình đa tầng:
 - Dễ dàng mở rộng, thay đổi quy mô của hệ thống
- Nhược điểm:
 - Truyền dữ liệu giữa các tầng chậm vì phải truyền giữa các tiến trình khác nhau
 - Phát triển ứng dụng khá phức tạp
→ áp dụng trong các hệ thống lớn



Khái niệm về mô hình đa tầng, đa lớp

Khái niệm về lớp

- Khác với tầng, lớp không thể hiện rõ sự tách biệt về mặt vật lý chúng thường nằm chung một nơi nhưng ở các namespace khác nhau.
- Mô hình lớp phân chia ứng dụng về mặt logic dựa vào chức năng.





Khái niệm về mô hình đa tầng, đa lớp

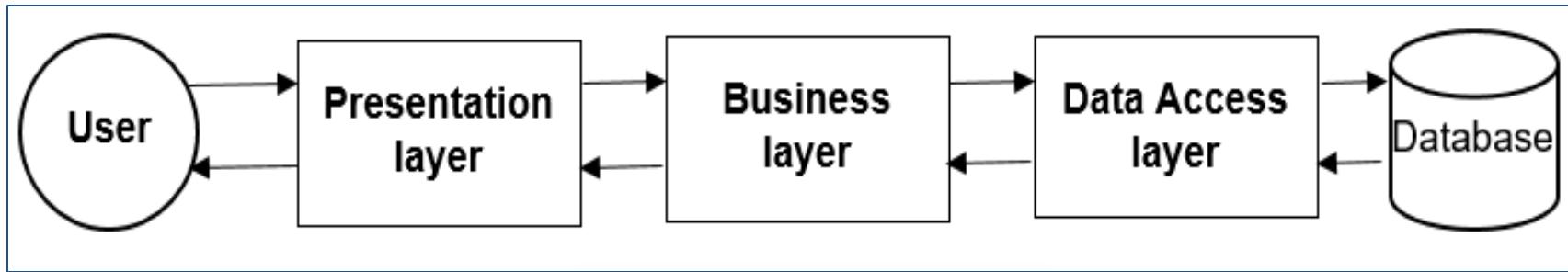
- Mô hình ba lớp (3-layer)

- Phân chia ứng dụng về mặt logic dựa vào chức năng.
- Các lớp khác nhau khi được thực thi có thể nằm trong cùng một vùng bộ nhớ của một tiến trình.
- Mô hình ba lớp bao gồm:
 - **Presentation Layer:** Thành phần giao diện, là các form của chương trình tương tác với người sử dụng.
 - **Business Layer:** Lớp xử lý các nghiệp vụ của chương trình như xử lý tính toán, xử lý ngoại lệ, an toàn dữ liệu,...
 - **Data Access Layer:** Lớp giao tiếp với các hệ quản trị cơ sở dữ liệu



Khái niệm về mô hình đa tầng, đa lớp

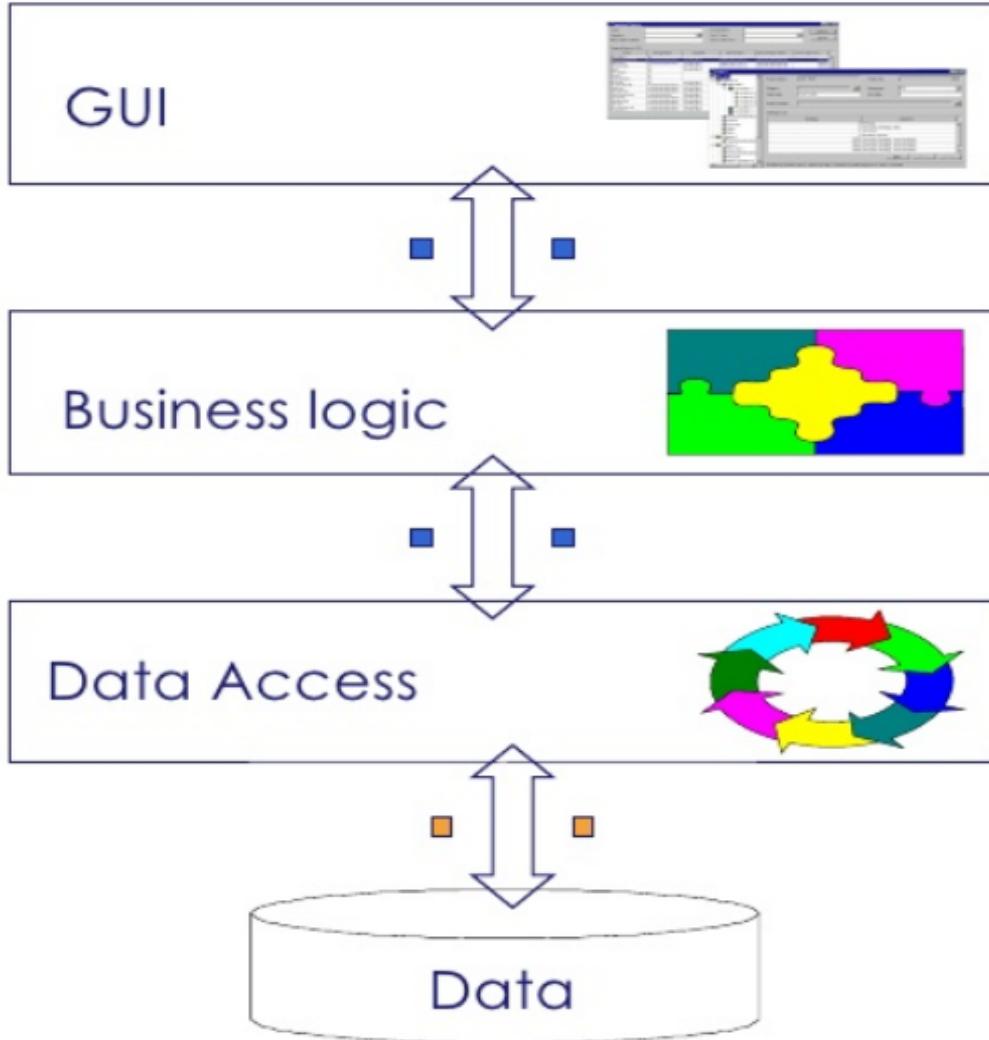
- Mô hình ba lớp (3-layer)



- Điểm khác nhau cơ bản giữa mô hình đa tầng và mô hình đa lớp?



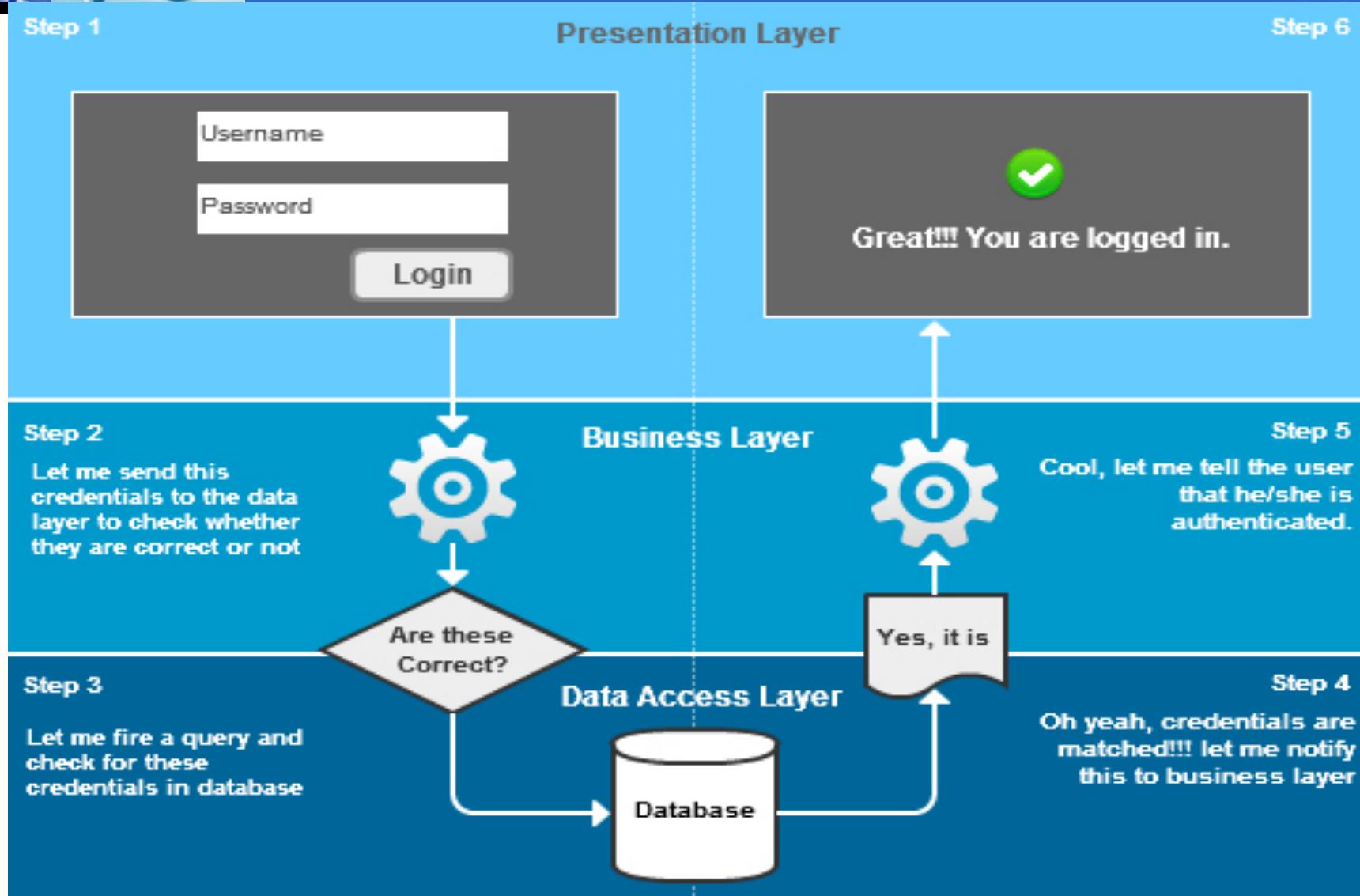
Việc liên lạc giữa các Layers



- Data Transfer Object (DTO)
- Các giá trị, dòng, bảng

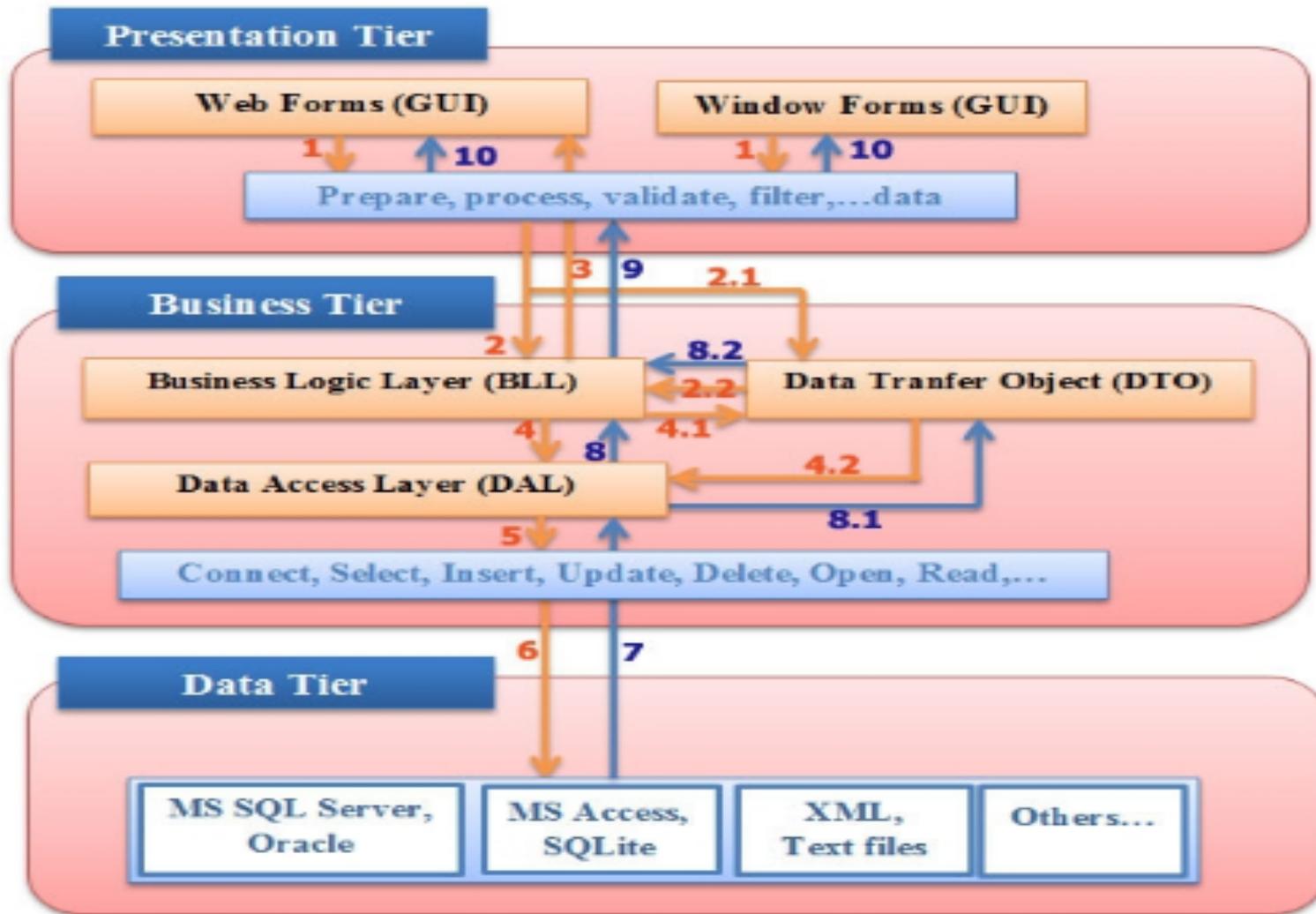


Ví dụ: Ứng dụng đa lớp





Cách thức hoạt động





Lợi ích mô hình đa lớp

- Phát triển phần mềm: có tính chuyên nghiệp hóa, có thể chia cho nhiều nhóm được đào tạo nhiều kỹ năng khác nhau.
- Bảo trì: các thành phần của một hệ thống dễ được thay đổi, không hoặc chỉ ảnh hưởng đến lớp ngay gần kề nó, không phải viết lại cả chương trình.
- Mở rộng: Thêm các chức năng mới cho từng lớp dễ dàng hơn



Các quy tắc cơ bản của giao diện phần mềm

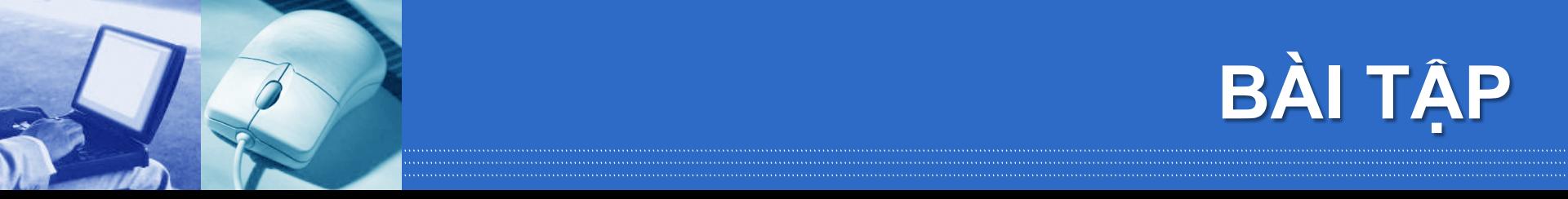
- Thận thiện: không gây cảm giác lúng túng, mất tự tin khi sử dụng chương trình. Mọi công cụ cần hiển thị rõ ràng có sắp xếp đúng thứ tự.
- Dễ sử dụng: Mọi danh mục, thanh công cụ, biểu tượng, văn bản minh họa được thiết kế gần gũi với nhận thức của người sử dụng.
- Tin cậy: Giao diện với màu sắc hài hòa, nội dung rõ ràng, mạch lạc, các thành phần chức năng được bố trí hợp lý.



Thiết kế chương trình đa lớp

- Bước 1: Xác định các yêu cầu, chức năng của ứng dụng.
- Bước 2: Phân loại và nhóm các chức năng của ứng dụng.
- Bước 3: Thiết kế cấu trúc dữ liệu, cơ sở dữ liệu.
- Bước 4: Xây dựng các lớp truy xuất dữ liệu.
- Bước 5: Thiết kế lớp ứng dụng: định nghĩa giao diện, đầu vào, đầu ra, cấu trúc dữ liệu.
- Bước 6: Xây dựng lớp giao diện tương tác với người dùng

- Thiết kế giao diện cho ứng dụng máy tính bỏ túi đơn giản.



BÀI TẬP

- Tạo một ứng dụng cơ bản tính tiền công dịch vụ cho một lần đi khám tại phòng nha.
- Với mỗi khách hàng, các dịch vụ cung cấp gồm: tẩy răng, cạo vôi, chụp hình răng và trám răng. Mỗi loại sẽ có chi phí riêng. Cuối cùng tính tổng các chi phí mà người khách phải trả. Lưu ý: chỉ tính tiền khi phần thông tin tên khách hàng đã được nhập (nếu thông tin này chưa có thì chương trình phát sinh MessageBox cảnh báo).



BÀI TẬP

Dental Payment Application

Dental Payment Form

Tên khách hàng |

<input type="checkbox"/> Cạo vôi	\$100.000
<input type="checkbox"/> Tẩy trắng	\$1.200.000
<input type="checkbox"/> Chụp hình răng	\$200.000
Trám răng	<input type="text" value="0"/> \$80.000/cái

Total

Thoát **Tính tiền**



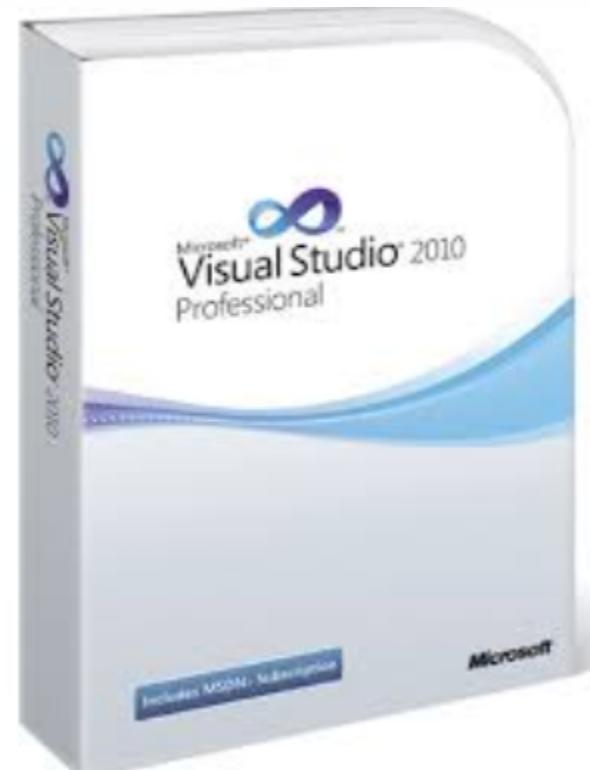
1.2. Giới thiệu công nghệ .NET của Microsoft

- Giới thiệu Microsoft Visual Studio.NET
- .NET Framework và Common Language Runtime
- Các ngôn ngữ trong Visual Studio.NET
- Các loại project trong Visual Studio.Net
- Tạo project mới trong Visual Studio
- Các thành phần trong một project
- Trang Start Page
- Minh họa xây dựng ứng dụng với Visual C#



1.2.1 Giới thiệu Microsoft Visual Studio.NET

- Công nghệ .NET của Microsoft được công bố vào năm 2000
- Cho phép tạo các ứng dụng .Net với nhiều loại ngôn ngữ như C#, Visual Basic, và những ngôn ngữ khác.
- Công nghệ ASP.NET cho phép tạo ra các ứng dụng và dịch vụ Web một cách trực quan.





Giới thiệu Microsoft Visual Studio.NET (tt)

- Các phiên bản của Visual Studio.NET
 - Visual Studio 2002 (12/2/2002): .NET Framework 1.0
 - Visual Studio 2003 (24/4/2003): .NET Framework 1.1
 - Visual Studio 2005 (10/2005): .NET Framework 2.0
 - Visual Studio 2008 (11/2007): .NET Framework 3.5
 - Visual Studio 2010 (12/4/2010): .NET Framework 4.0
 - Visual Studio 2012 (15/8/2012): .NET Framework 4.5
 - Visual Studio 2013: .NET Framework 4.5.2, hỗ trợ các công nghệ mới như Windows Phone, Windows Store,...
 - Visual Studio 2015: .NET Framework 4.6, hỗ trợ đa nền tảng, cho phép xây dựng ứng dụng chạy trên hệ điều hành Linux, Mac OS, lập trình ứng dụng cho Android, iOS.
 - Visual Studio 2017: .NET Framework 4.7
 - Visual Studio 2019: .NET Framework 5.0



Giới thiệu Microsoft Visual Studio.NET (tt)

- Mỗi phiên bản Visual Studio.NET tương thích với một version .NET Framework.
- Ví dụ, một chương trình được viết với Visual Studio 2015 chỉ có thể chạy được trên hệ điều hành có cài .NET Framework 4.6.
- Các phiên bản Visual Studio .NET của Microsoft liên tục được cập nhật và phát triển.



1.2.2 .NET Framework và Common Language Runtime

- .NET Framework là một thành phần quan trọng trong công nghệ .NET, là nền tảng cho phép phát triển các loại ứng dụng:
 - Desktop: Windows Form Application, ứng dụng văn phòng (office/sharepoint), Windows Presentation Foundation (WPF).
 - Windows Store: các ứng dụng dành cho thiết bị cảm ứng.
 - Web: các ứng dụng web (ASP.NET), dịch vụ web (Web Service), Silverlight.
 - Windows Communication Foundation(WCF) services: các ứng dụng dạng dịch vụ chia sẻ dữ liệu qua mạng, internet...

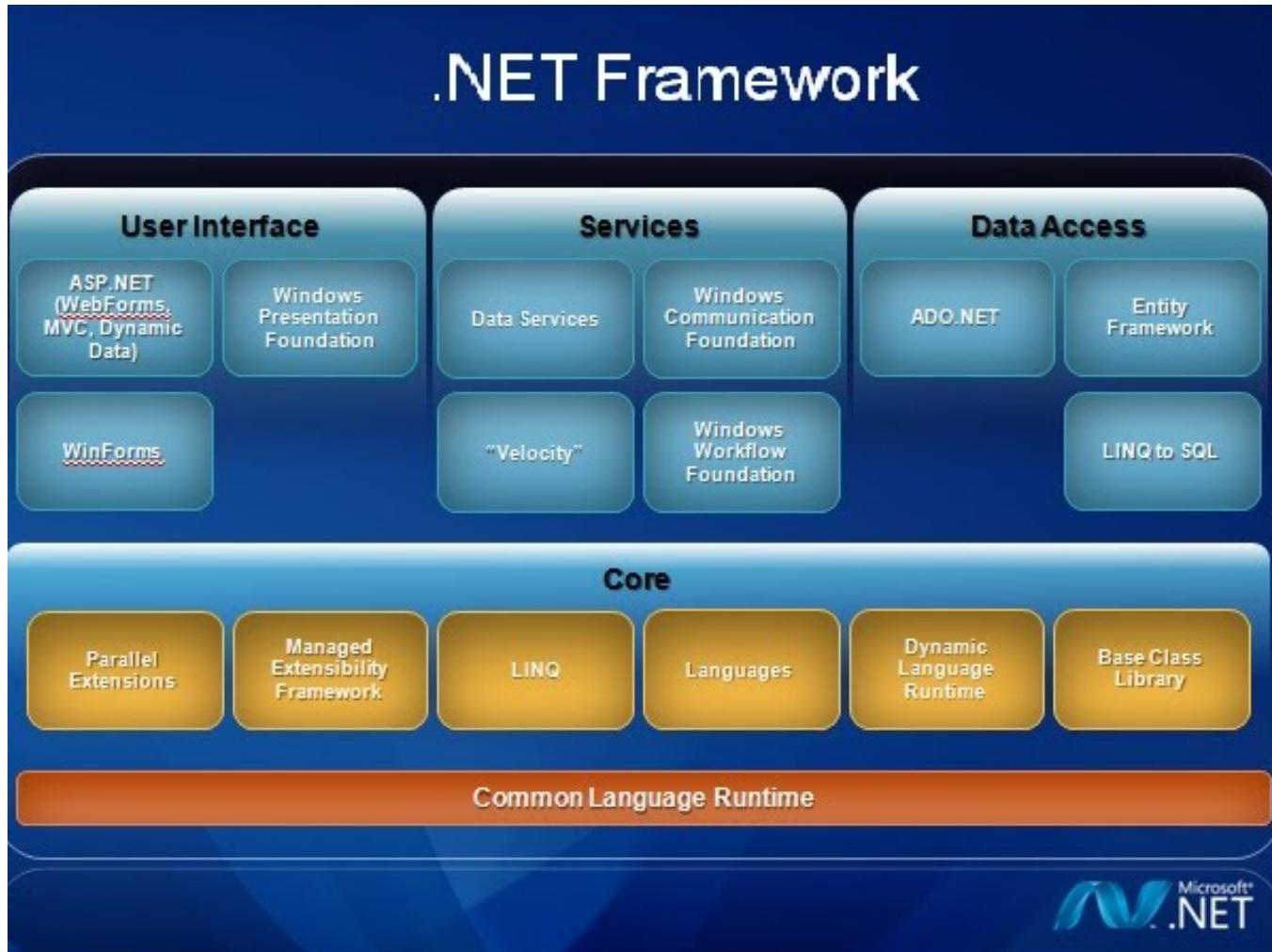


.NET Framework và Common Language Runtime (tt)

- Bao gồm hai thành phần chính:
 - Lớp thư viện Framework (Framework Class Library)
 - Bộ thực thi ngôn ngữ chung (CommonLanguageRuntime).



.NET Framework và Common Language Runtime (tt)





Framework Class Library

- Là thư viện chứa các lớp cho phép xây dựng các loại ứng dụng với các ngôn ngữ khác nhau.
 - Base class library: thư viện các lớp cơ bản nhất, ví dụ: String, Integer, Exception, ...
 - ADO.NET và XML: thư viện các lớp dùng để xử lý dữ liệu.
 - ASP.NET: thư viện các lớp dùng để xây dựng ứng dụng web.
 - Web services: thư viện các lớp dùng để xây dựng các dịch vụ được cung cấp qua mạng và internet.
 - Windows Form: thư viện các lớp đối tượng dành cho việc xây dựng các ứng dụng có giao diện đồ họa.



Common Language Runtime (CLR)

- Là thành phần kết nối giữa các phần trong .NET Framework với hệ điều hành.
- Các thành phần bên trong CLR:
 - Class loader
 - Just In Time compiler
 - Garbage collector
 - Exception handler
 - COM marshaller
 - Security engine
- Các chức năng của CLR:
 - Bảo vệ bộ nhớ
 - Xử lý lỗi
 - Thu gom rác tự động
 - Kiểm tra tính an toàn, không cho phép các lệnh nguy hiểm được thi hành.



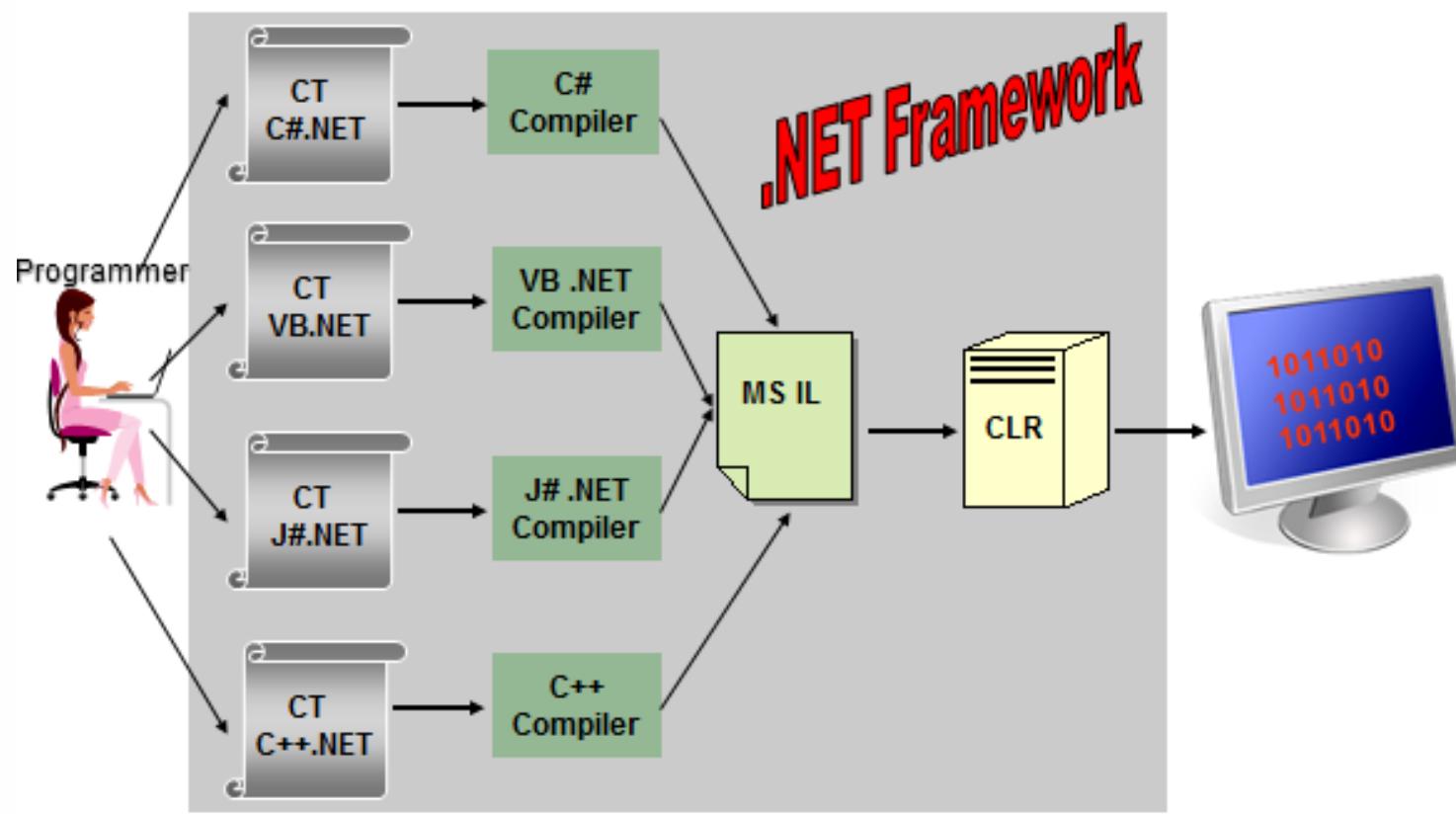
Common Language Runtime (CLR)

- Máy tính cài đặt .NET Framework có thể thực thi một chương trình .NET có version phù hợp.
- Khả năng tương tác giữa các ngôn ngữ:
 - Chương trình có thể được viết bằng ngôn ngữ khác nhau (C#, Visual Basic,...)
→ biên dịch thành MSIL.
→ .NET Framework trở thành một ngôn ngữ độc lập, không gắn với một ngôn ngữ lập trình cụ thể.



Common Language Runtime (CLR)

- Quá trình biên dịch và thực thi một chương trình .NET





1.2.3 Các ngôn ngữ trong VS

- Visual C#: ngôn ngữ mới, hỗ trợ lập trình trực quan, hướng sự kiện, cú pháp tương tự C++.
- Visual Basic: được phát triển từ BASIC, một ngôn ngữ lập trình đơn giản, dành cho người mới học về kỹ thuật lập trình cơ bản, hỗ trợ lập trình trực quan, hướng sự kiện, chỉ có trong.NET.
- Visual C++: nền tảng là C++, cho phép phát triển các ứng dụng với thư viện MFC, .NET.
- Visual F#: ngôn ngữ lập trình hàm dành cho nhà phát triển .NET, mới được tích hợp vào Visual Studio.
- Python: ngôn ngữ lập trình dạng script.



1.2.4 Các loại project trong Visual Studio .Net

- Windows Forms Application: ứng dụng dạng giao diện đồ họa.
- Class Library: tạo các thư viện liên kết động dùng trong các ứng dụng khác.
- ASP .NET Web Application / Service / Control Library: các ứng dụng về web, dịch vụ web.
- Console Application: ứng dụng dạng giao diện dòng lệnh.
- Windows Service: các ứng dụng cung cấp dịch vụ chạy trên máy server.
- Windows Presentation (WPF): ứng dụng giao diện đồ họa trên nền tảng XAML.



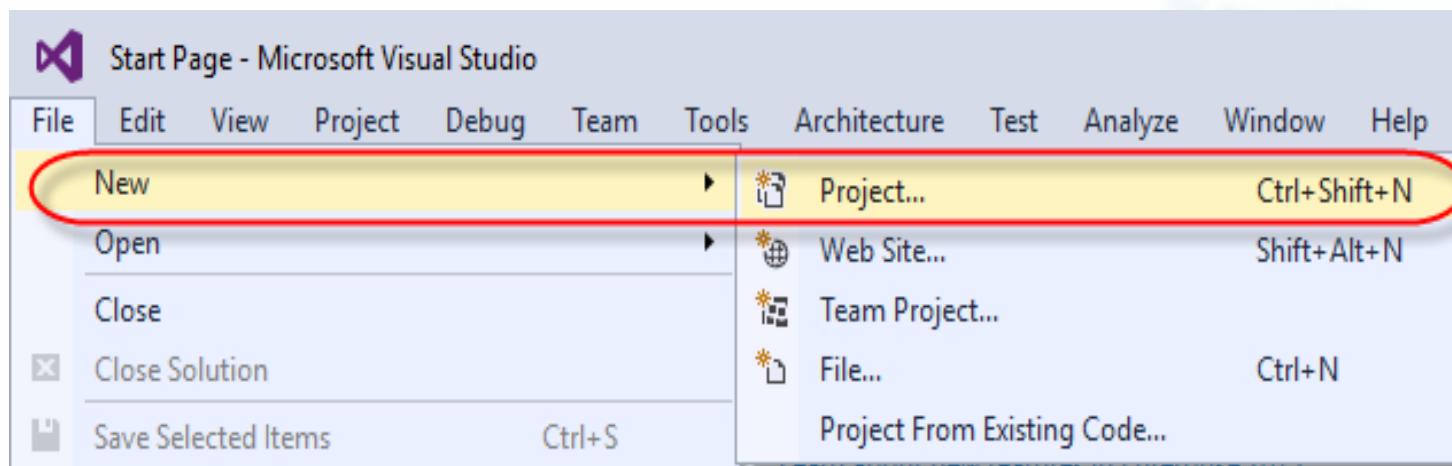
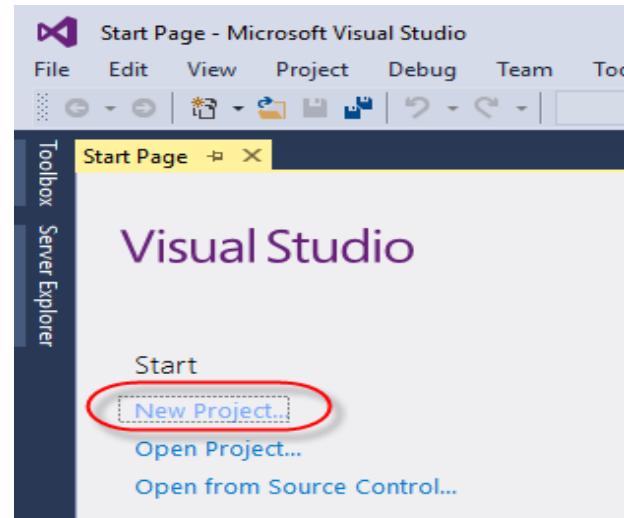
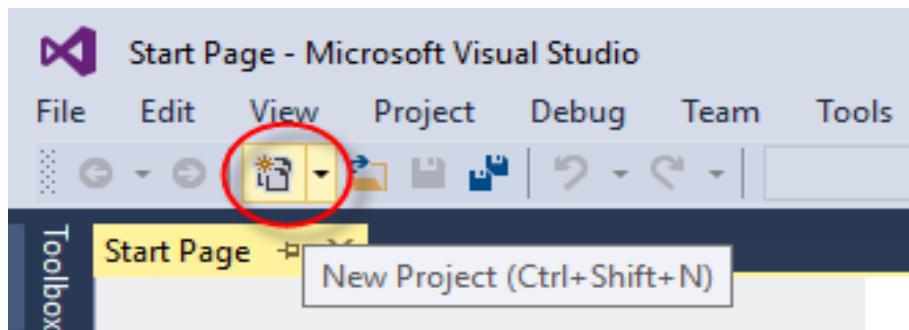
Các loại project trong Visual Studio.Net (tt)

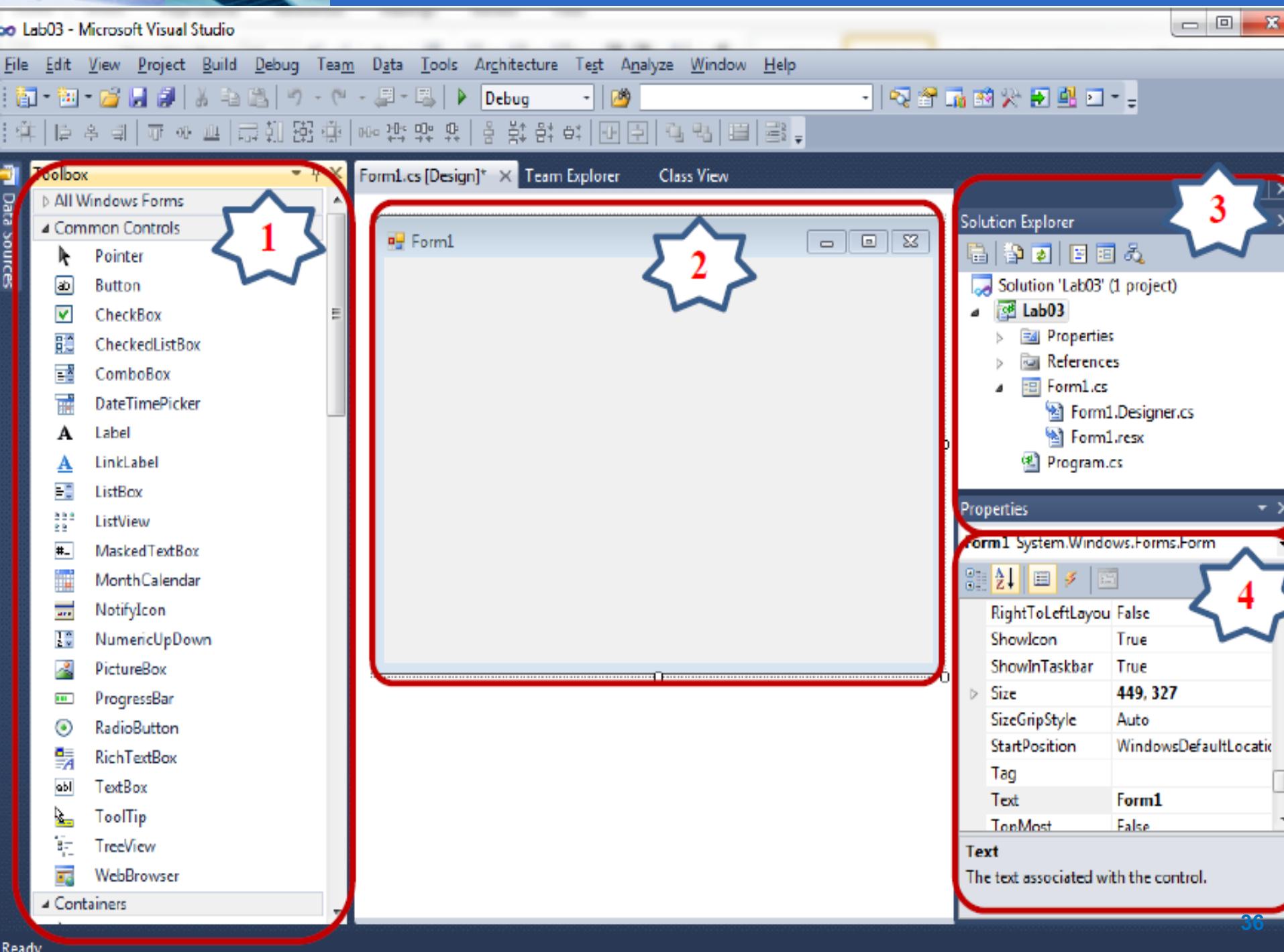
- Windows Communication Foundation
- Application (WCF): các ứng dụng hướng dịch vụ trên nền Windows.
- Silverlight Application, Silverlight Class LibraryOthers: ứng dụng đa phương tiện chạy trên nền web.
- Excel WorkBook, Word Document: các ứng dụng văn phòng tương tự Microsoft Word, Excel.
- Cloud: các ứng dụng điện toán đám mây.
- Android, IOS, Windows Phone: ứng dụng trên thiết bị di động.



1.2.5 Tạo project mới trong Visual Studio

- Ba cách:







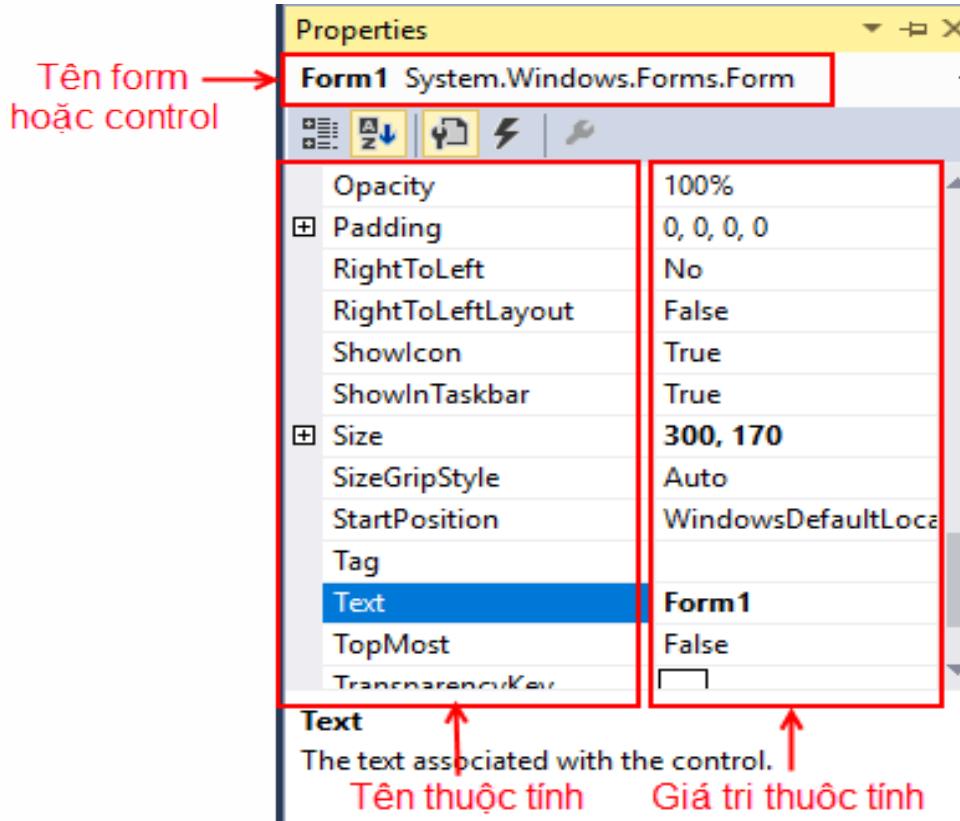
Các thành phần trong một project (tt)

- Form1: cửa sổ chính của ứng dụng
- Cửa sổ Solution Explorer: chứa các thành phần có trong project, bao gồm Project References, Forms, Classes, Modules, Forder với các items
- Error List: hiển thị kết quả biên dịch và thông báo lỗi (nếu có)



Các thành phần trong một project (tt)

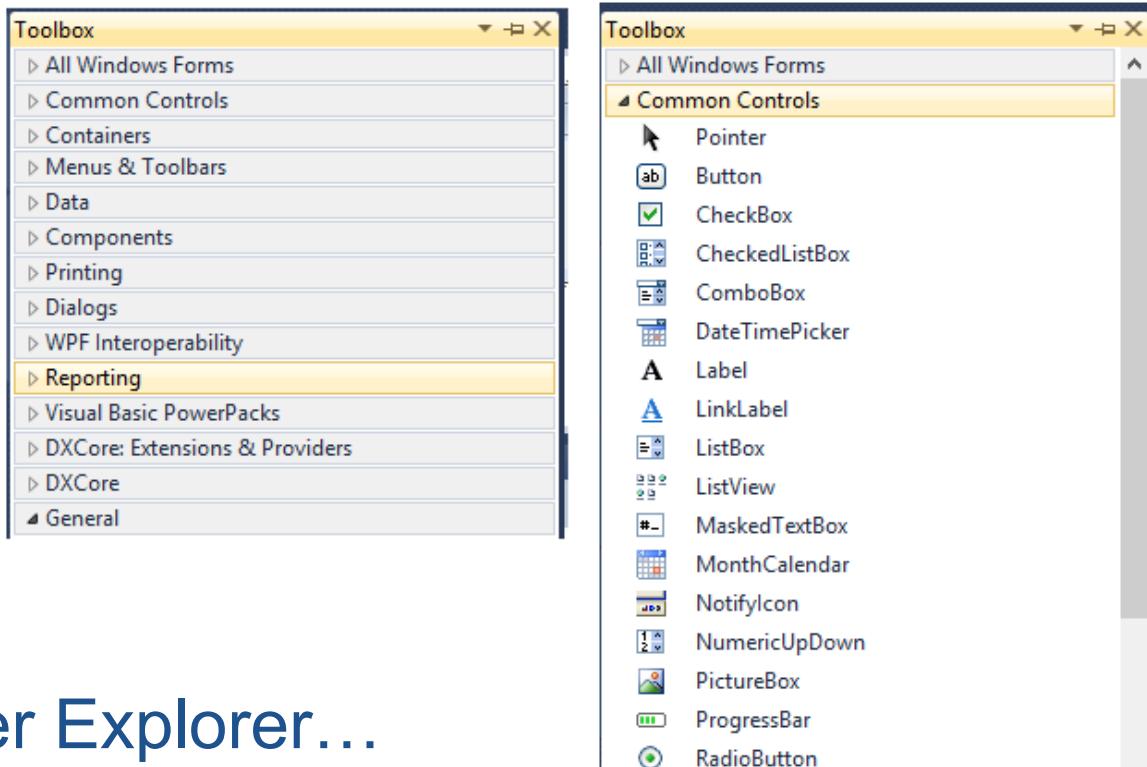
- Cửa sổ Properties: cho phép thiết lập các thuộc tính cho Form và controls





Các thành phần trong một project (tt)

- Thanh Toolbox:

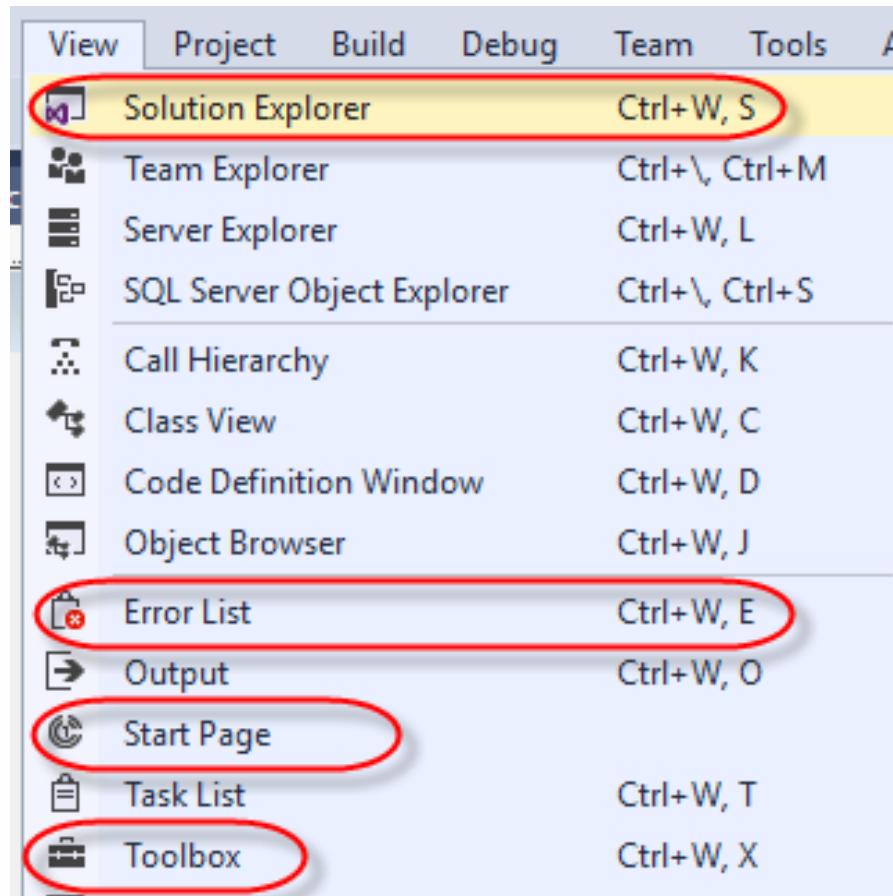


- Cửa sổ Server Explorer...



Các thành phần trong một project (tt)

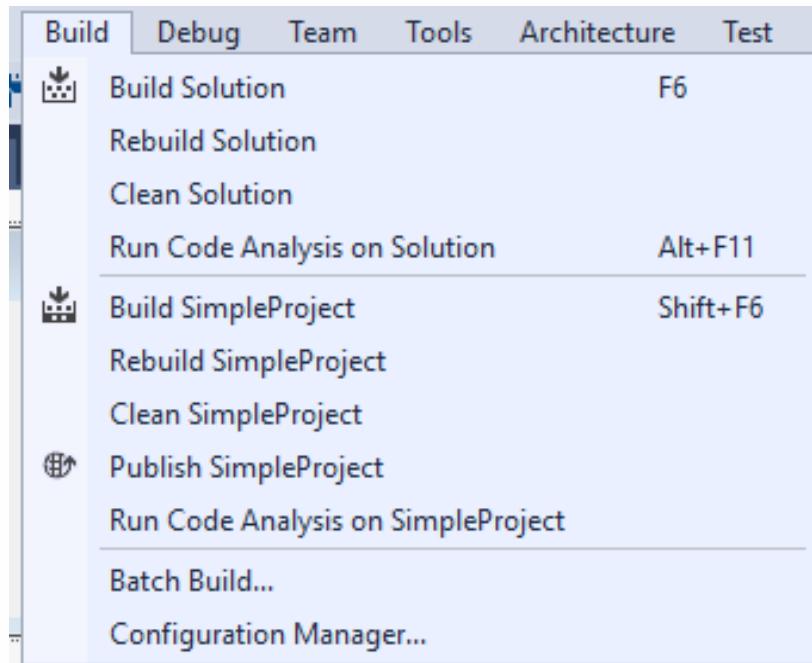
- Hiển thị/ ẩn các thành phần: menu View





Các thành phần trong một project (tt)

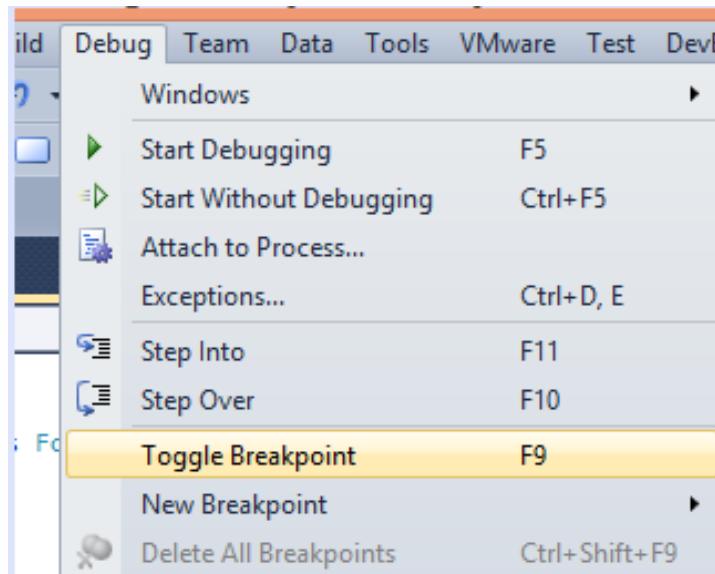
- Biên dịch: menu Build
 - Build Solution / Rebuild Solution
 - Build Project-name /Rebuild Project-name:





Các thành phần trong một project (tt)

- Debug (chạy từng bước):
 - Chạy chương trình theo từng bước để theo dõi giá trị các biến, các đối tượng, được sử dụng rất hiệu quả trong trường hợp chương trình bị lỗi logic, cho kết quả không đúng.

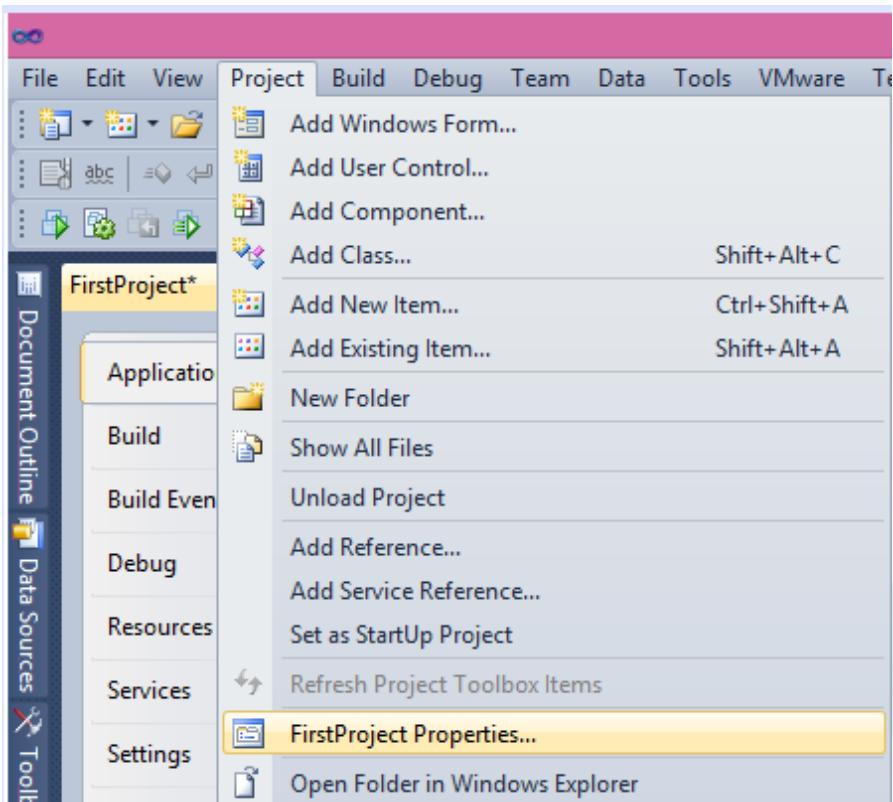




Các thành phần trong một project (tt)

- Thiết lập các thuộc tính cho Project:
 - Từ menu Project, chọn Project-name Properties

- hoặc click project Solution
chọn Properties





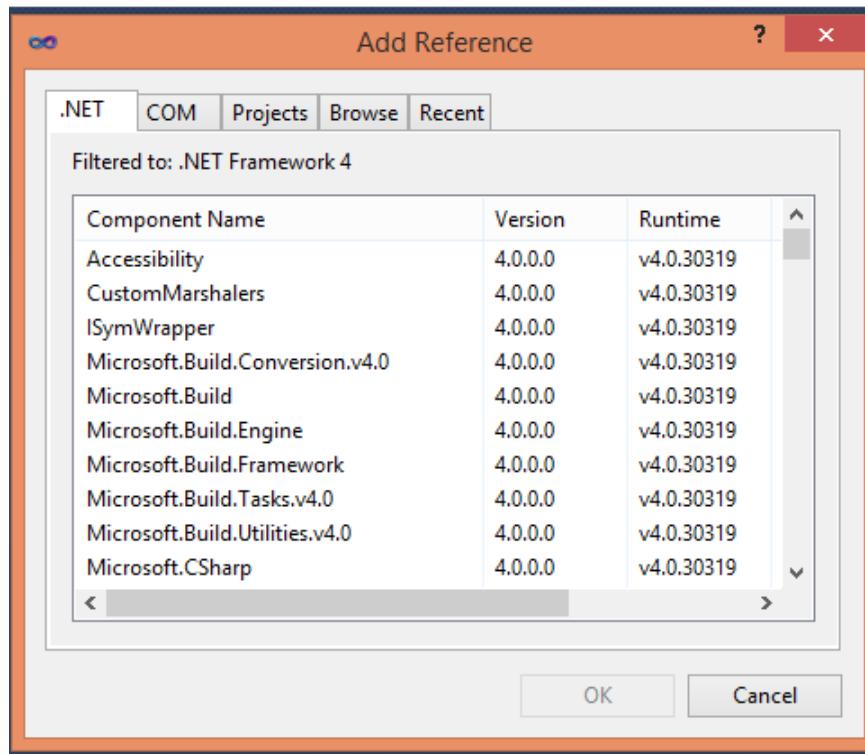
Các thành phần trong một project (tt)

- Thiết lập các thuộc tính cho Project:
 - Thiết lập các thuộc tính thông thường
 - Định nghĩa tên Assembly
 - Namespace gốc
 - Đầu ra Project
 - Đối tượng chạy đầu tiên
 - Import các mức namespaces
 - Thiết lập các thuộc tính cấu hình
 - Debug
 - Các tùy chọn Build



Các thành phần trong một project (tt)

- Thiết lập các tham chiếu trong Project:
 - Sử dụng hộp thoại Add Reference





1.2.7 Trang Start Page

- Chứa danh sách các liên kết đến các nguồn tài nguyên Visual Studio IDE, mỗi liên kết cho phép truy cập đến thông tin có liên quan từ các web site.
Bao gồm:
 - Phần Start: chứa các nút cho phép tạo hoặc mở project.
 - Phần Recent: chứa thông tin về các project mới nhất được tạo hoặc sửa đổi giúp mở nhanh các project này.
 - Các liên kết khác: chứa các thông tin giới thiệu và hướng dẫn về cách sử dụng các tài nguyên trong .NET.

Trang Start Page (tt)



Start Page Form1.cs* Form1.cs [Design]*

Visual Studio

Start

- New Project...
- Open Project...
- Open from Source Control...

Recent

- FirstProject
- website1(2)
- Setup1
- WindowsFormsApplication2
- Protect_Word
- WordDocument1
- 1554050114_NguyenDoanKieuVy_C06
- 1451010214_PhanThiTuongVi_C05
- 1554050112_NguyenThiThuyVan_C04
- 1554050111_VOHAYUEN_03

Discover what's new in Enterprise 2015

Learn about new features in Enterprise 2015
See what's new in the .NET Framework
Explore what's new in Visual Studio Team Services

Ready to Cloud-power your experience?

Connect to Azure →

News

[7 lesser known hacks for debugging in Visual Studio](#)
The Visual Studio debugger is a magical beast that can save you loads of time while finding and fixing issues in your application. It is chock-full of tools that can mak...
NEW Thứ Sáu, Tháng Sáu 30, 2017

[Smooth as Butter Animations in the Visual Layer with the Windows 10 Creators Update](#)
The Windows 10 Creators Update marks the third major release of the Windows UI platform APIs. With each release, an attempt is frequently made to simplify feat...
NEW Thứ Tư, Tháng Sáu 28, 2017

[Going Global with Xamarin and Azure Cosmos DB](#)
Imagine that MyTeaCompany is a locally renowned business in Pune, India, supplying tea to the area from their ten shops within the city. They supply a huge va...
NEW Thứ Sáu, Tháng Sáu 30, 2017

[Windows Template Studio 1.1 released!](#)
We're extremely excited to announce the Windows Template Studio 1.1. In partnership with the community, we've been cranking out and iterating ne...
NEW Thứ Tư, Tháng Sáu 28, 2017

New on Microsoft Platforms

- Windows
- Microsoft Azure
- ASP.NET Core and Web
- Microsoft Office
- SharePoint Development

Featured Videos

-  IntelliTrace in Visual Studio 2015 12:13
-  Windows Presentation Foundation (WPF) Application Development 15:49

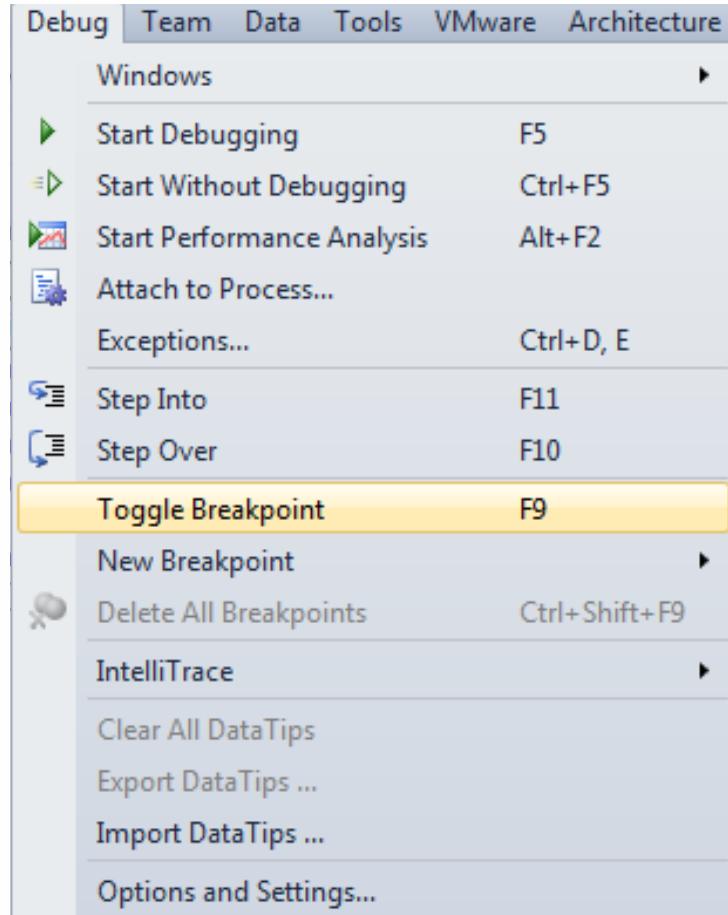


How to Debugging?





Debug chương trình



1. Chọn dòng text
2. Debug menu
3. Chọn Toggle Breakpoint hoặc nhấn F9



Debug chương trình

Program.cs X

project3.Program

Main(str

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

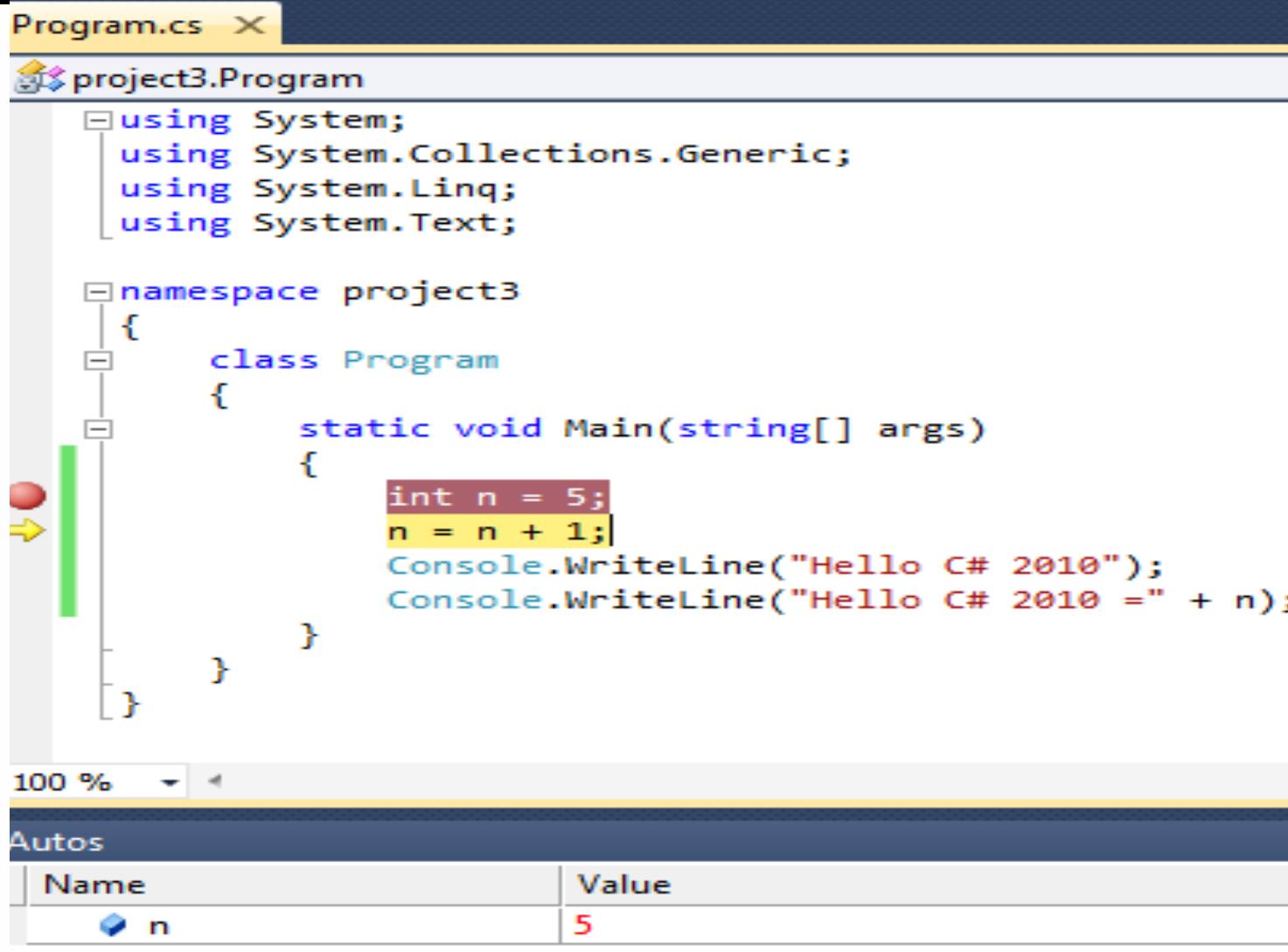
namespace project3
{
    class Program
    {
        static void Main(string[] args)
        {
            int n = 5;
            n = n + 1;
            Console.WriteLine("Hello C# 2010");
            Console.WriteLine("Hello C# 2010 =" + n);
        }
    }
}
```

Breakpoint here





Debug chương trình



The screenshot shows a debugger interface with the following details:

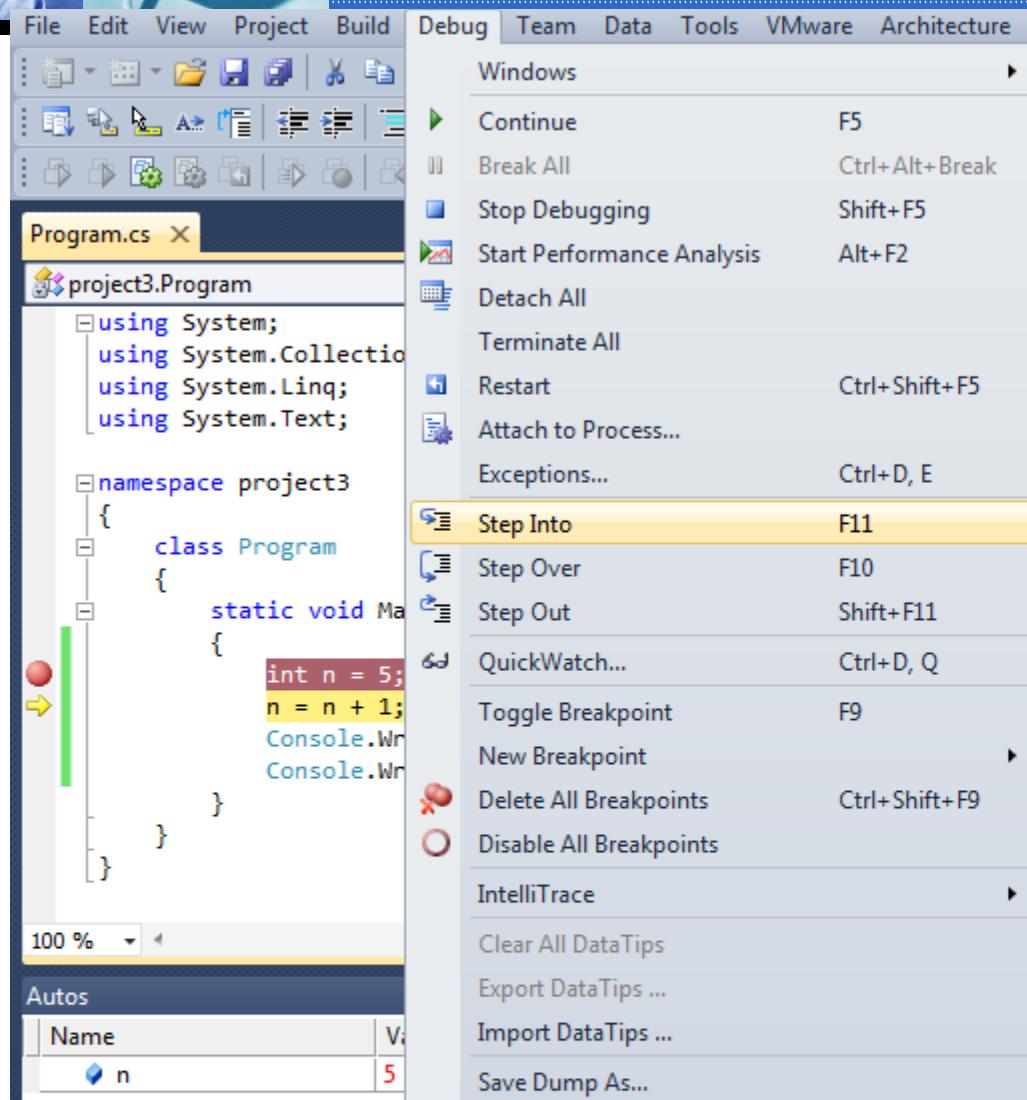
- Code Editor:** The file `Program.cs` is open, displaying the following C# code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace project3
{
    class Program
    {
        static void Main(string[] args)
        {
            int n = 5;
            n = n + 1;
            Console.WriteLine("Hello C# 2010");
            Console.WriteLine("Hello C# 2010 =" + n);
        }
    }
}
```
- Breakpoint:** A red circular breakpoint icon is visible on the left margin next to the first line of code.
- Execution Status:** A green vertical bar indicates the current execution point is at the line `n = n + 1;`.
- Autos Window:** The bottom window displays the variable `n` with the value `5`.



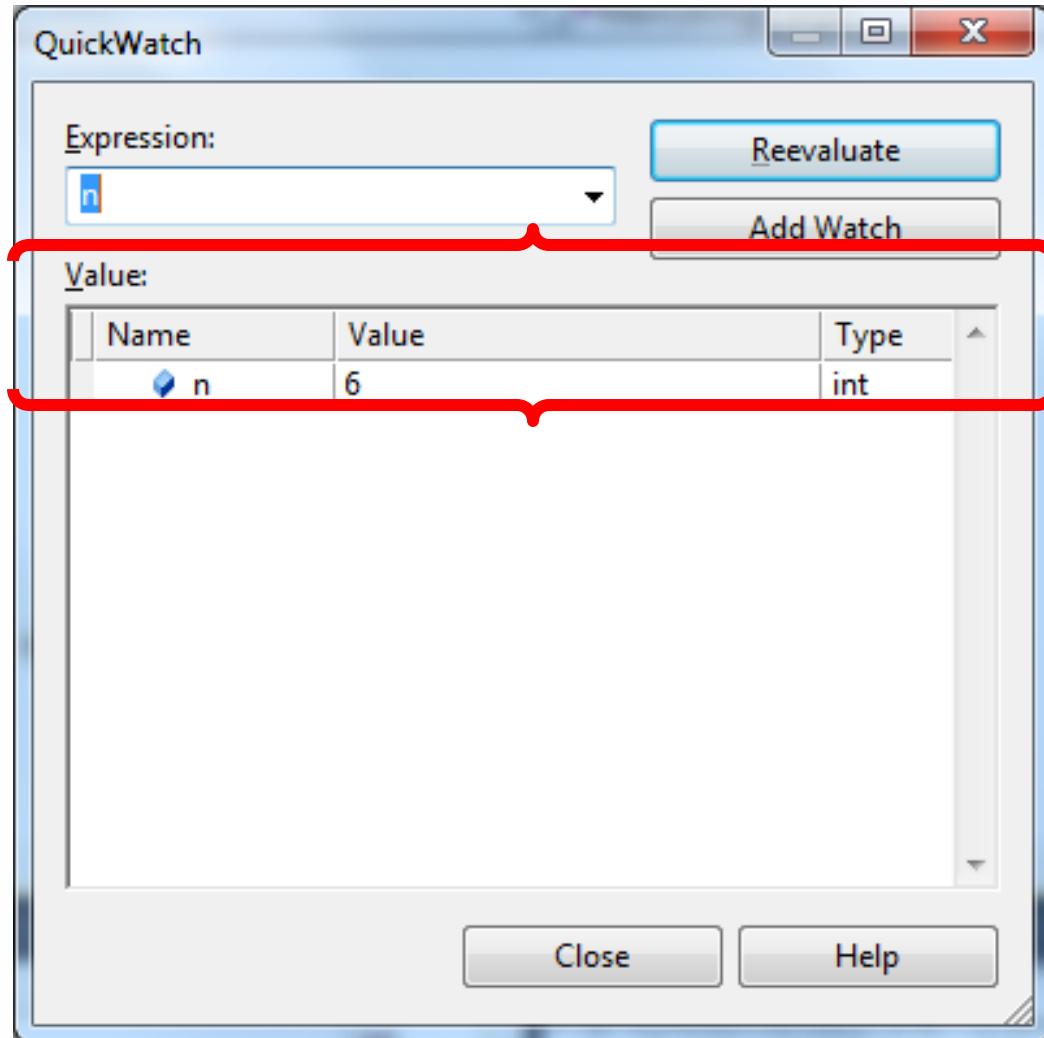
Debug chương trình



Step Into
Step Over
Step Out



Debug chương trình





File Edit View Project Build Debug Team Data Tools VMware

Program.cs X

project3.Program

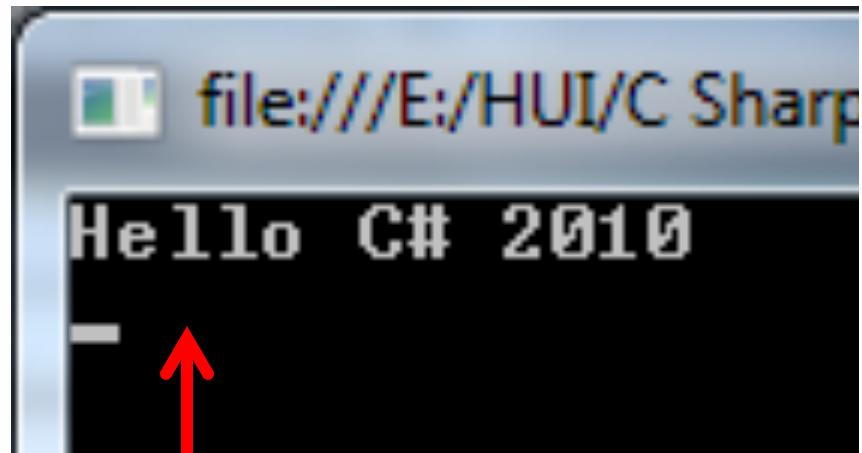
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace project3
{
    class Program
    {
        static void Main(string[] args)
        {
            int n = 5;
            n = n + 1;
            Console.WriteLine("Hello C# 2010");
            Console.WriteLine("Hello C# 2010 =" + n);
        }
    }
}
```

100 %

Watch 1

Name	Value
n	10





1.2.8 Minh họa xây dựng ứng dụng với Visual C#

- Tạo ứng dụng
- Đóng gói ứng dụng

