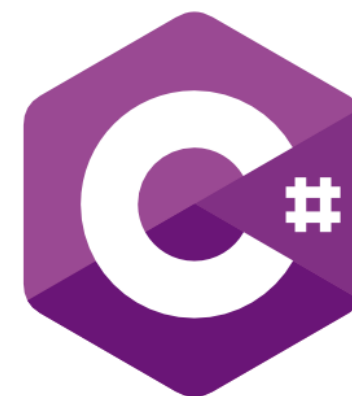


C# 1 – Bắt đầu với C#

Giảng viên: **ThS. Lê Thiện Nhật Quang**
Email: quangln.dotnet.vn@gmail.com
Website: <http://dotnet.edu.vn>
Điện thoại: **0868.917.786**

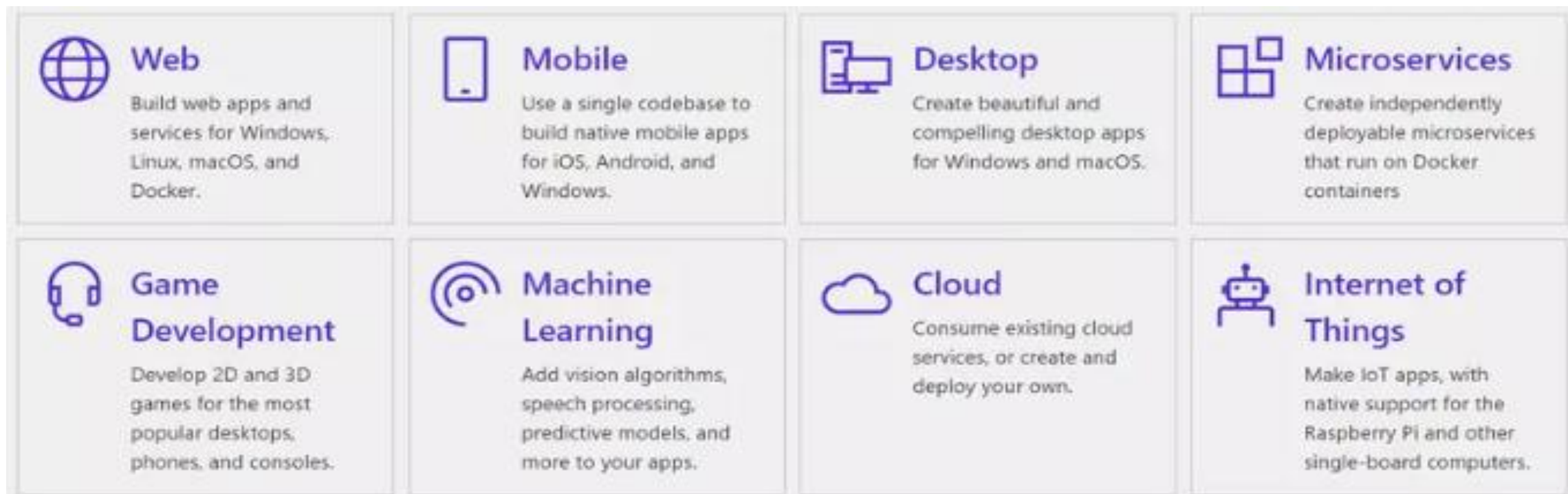


MỤC TIÊU

- Tìm hiểu .NET Framework
- Tìm hiểu tính năng của C#
- Làm quen phần mềm Visual Studio, các Plugin
- Khuyến khích sử dụng phần mềm Visual Studio 2022

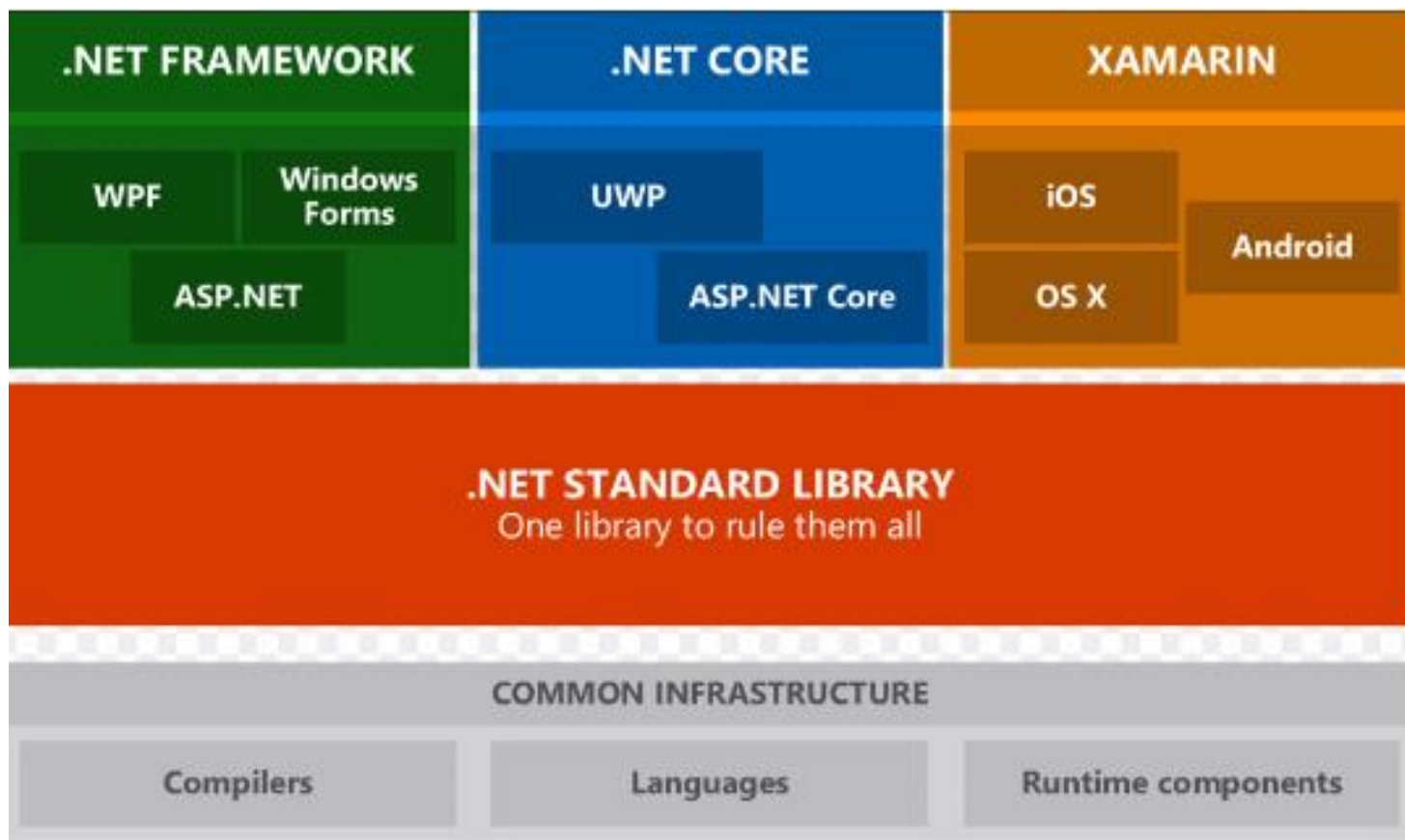
1.1. GIỚI THIỆU .NET

- .Net là nền tảng phát triển mã nguồn mở (open source), đa nền tảng (cross-platform), miễn phí.
- Các ngôn ngữ .Net (như: C#, F#, hoặc Visual Basic), editors(như: Visual studio, Visual Studio Code...), các thư viện để viết các ứng dụng .Net (như: web, mobile, desktop, games, và IoT...) và một số ứng dụng .Net là đa nền tảng, một số ứng dụng thì hoạt động trên hệ điều hành hoặc.NET implementations (tạm dịch là thực thi của .Net). Cụ thể:



1.2. GIỚI THIỆU .NET

- Cho phép xây dựng, triển khai và chạy ứng dụng khác nhau và các dịch vụ sử dụng công nghệ .NET
- Sử dụng .Net framework giảm thiểu tối đa thời gian phát triển và triển khai phần mềm
- Giảm thiểu xung đột khi người sử dụng triển khai các hệ thống trên nhiều phiên bản khác nhau.

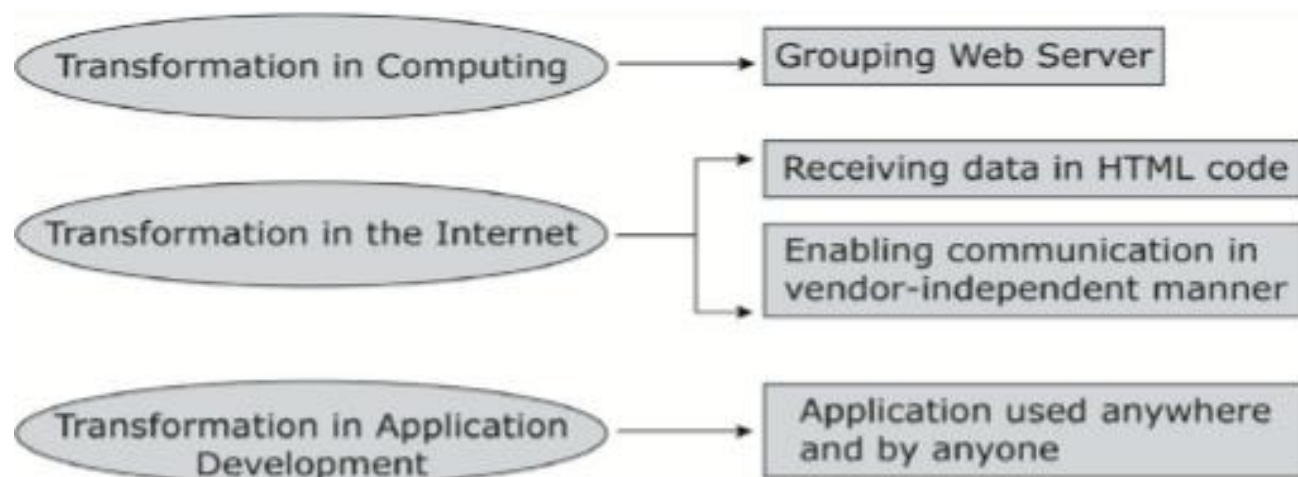


1.3. KIẾN TRÚC .NET

Với những cải tiến trong công nghệ mạng:

- Distributed computing- **Điện toán phân tán** đã cung cấp cách sử dụng hiệu quả sức mạnh xử lý của client và server.
- Ngoài ra, với sự xuất hiện của internet, ứng dụng đã trở thành một nền tảng độc lập, đảm bảo rằng ứng dụng đó có thể chạy trên nhiều PC với nhiều phần cứng và phần mềm kết hợp với nhau.
- Tương tự, với sự chuyển đổi trong phát triển ứng dụng, nó đã có thể cho phép clients và server giao tiếp với nhau một cách độc lập qua một **nhà cung cấp độc lập**.

1.3. KIẾN TRÚC .NET(2)



Hình trên cho thấy các tính năng khác nhau kèm theo những biến đổi trong điện toán, Internet, và phát triển ứng dụng.

Tất cả những chuyển đổi này được hỗ trợ bởi nền tảng công nghệ cung cấp bởi Microsoft gọi là .NET Framework. Dữ liệu được lưu trữ bằng .NET Framework có thể truy cập đến người dùng ở **bất cứ nơi đâu, bất cứ lúc nào**, thông qua .NET compatible device.

1.3. KIẾN TRÚC .NET (3)

.NET Framework là một nền tảng lập trình sử dụng để phát triển Windows, Web-based, và ứng dụng điện thoại. Nó là một giải pháp mã hóa để quản lý các chương trình được viết trên framework.

Nền tảng .NET Framework dựa trên 2 công nghệ cơ bản cho truyền tải dữ liệu:

- Extensible Markup Language(XML)
- Các bộ giao thức Internet

Các tính năng chính của XML:

- Phân chia riêng biệt giữa dữ liệu thực tế và cách trình bày
- Cho phép thông tin có thể được tổ chức, lập trình, và chỉnh sửa
- Cho phép trang Web cộng tác và cung cấp các nhóm dịch vụ Web
- Cung cấp giải pháp để dữ liệu được phân phối cho một loạt các thiết bị

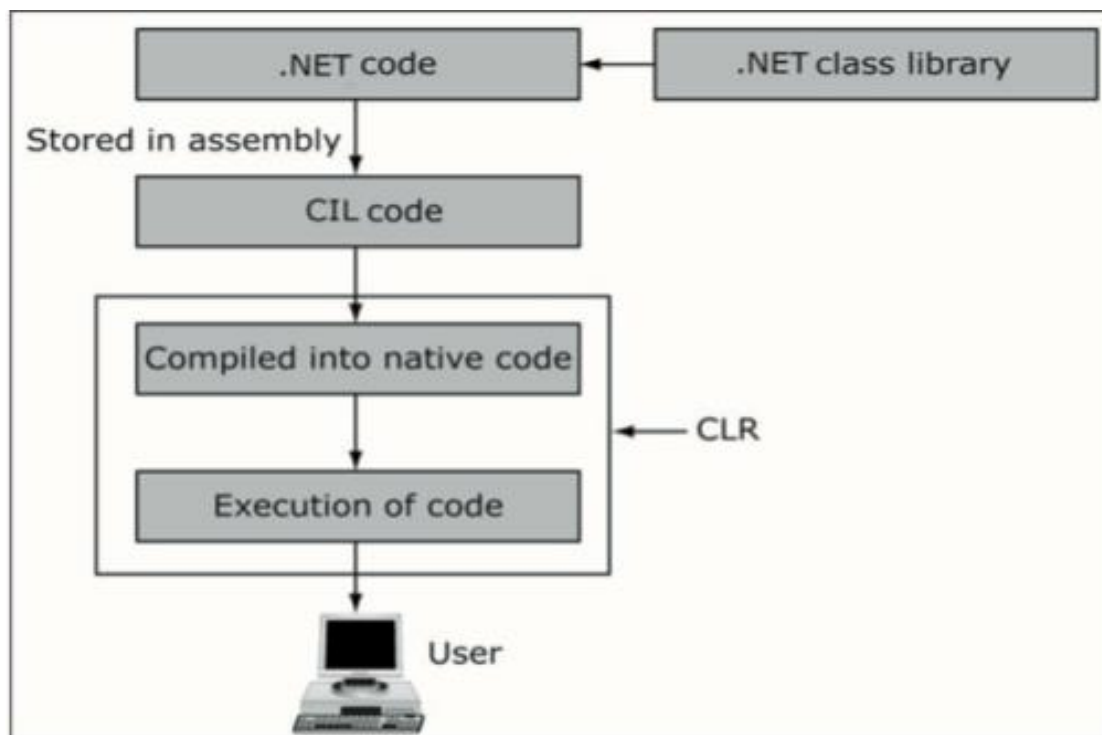
Ngoài XML, nền tảng .NET cũng được xây dựng trên **Giao thức Internet** như

- Hypertext Transfer Protocol (HTTP)
- Open Data Protocol (OData)
- Simple Object Access Protocol (SOAP)

1.3. KIẾN TRÚC .NET (4)

Trong các ứng dụng Windows truyền thống, codes trực tiếp biên soạn thành mã gốc thực thi của hệ điều hành.

Tuy nhiên, sử dụng .NET Framework: code của chương trình sẽ được biên dịch bằng CIL(trước đây gọi là MSIL) và được lưu vào 1 file gọi là assembly. Assembly sau đó được biên dịch bởi Common language Runtime (CLR) với mã nguồn gốc tại run-time



1.3. KIẾN TRÚC .NET (5)

Common Language Runtime (CLR): là trình thực thi để xử lý các ứng dụng đang chạy. Nó cung cấp các dịch vụ như

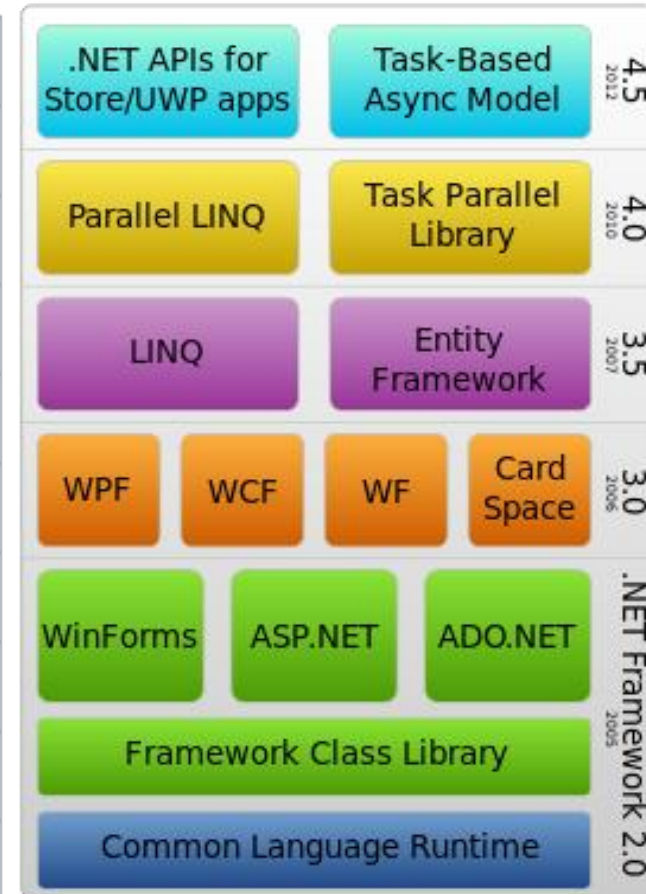
1. Quản lý bộ nhớ
2. Thực thi mã
3. Xử lý ngoại lệ
4. Kiểm định mã an toàn
5. Garbage Collectors: quá trình tự động thực thi nhiệm vụ quản lý bộ nhớ

Class Library: cung cấp một tập các API và các kiểu dữ liệu cho các chức năng thông thường. Nó cung cấp các kiểu chuỗi, ngày tháng, số, ... Class Library bao gồm các API để đọc và ghi file, kết nối với cơ sở dữ liệu, vẽ và nhiều thứ khác nữa.

1.3. KIẾN TRÚC .NET (6)

Lịch sử phát triển của .NET

| Phiên bản | Số hiệu phiên bản | Ngày phát hành | Visual Studio | Được phát hành kèm theo |
|-----------|-------------------|----------------------|------------------------|--|
| 1.0 | 1.0.3705.0 | 13 tháng 2 năm 2002 | Visual Studio.NET | Windows XP Tablet and Media Center Editions ^[2] |
| 1.1 | 1.1.4322.573 | 24 tháng 4 năm 2003 | Visual Studio.NET 2003 | Windows Server 2003 |
| 2.0 | 2.0.50727.42 | 7 tháng 11 năm 2005 | Visual Studio 2005 | Windows Server 2003 R2 |
| 3.0 | 3.0.4506.30 | 6 tháng 11 năm 2006 | | Windows Vista, Windows Server 2008 |
| 3.5 | 3.5.21022.8 | 19 tháng 11 năm 2007 | Visual Studio 2008 | Windows 7, Windows Server 2008 R2 |
| 4.0 | 4.0.30319.1 | 12 tháng 4 năm 2010 | Visual Studio 2010 | |
| 4.5 | 4.5.50709 | 15 tháng 8 năm 2012 | Visual Studio 2012 | Windows 8, Windows Server 2012 |
| 4.6 | | 20 tháng 7 năm 2015 | Visual Studio 2015 | Windows 10 |
| 4.8 | | 18 tháng 4 năm 2019 | | Windows 10 (ngừng phát triển từ bản 4.8) |



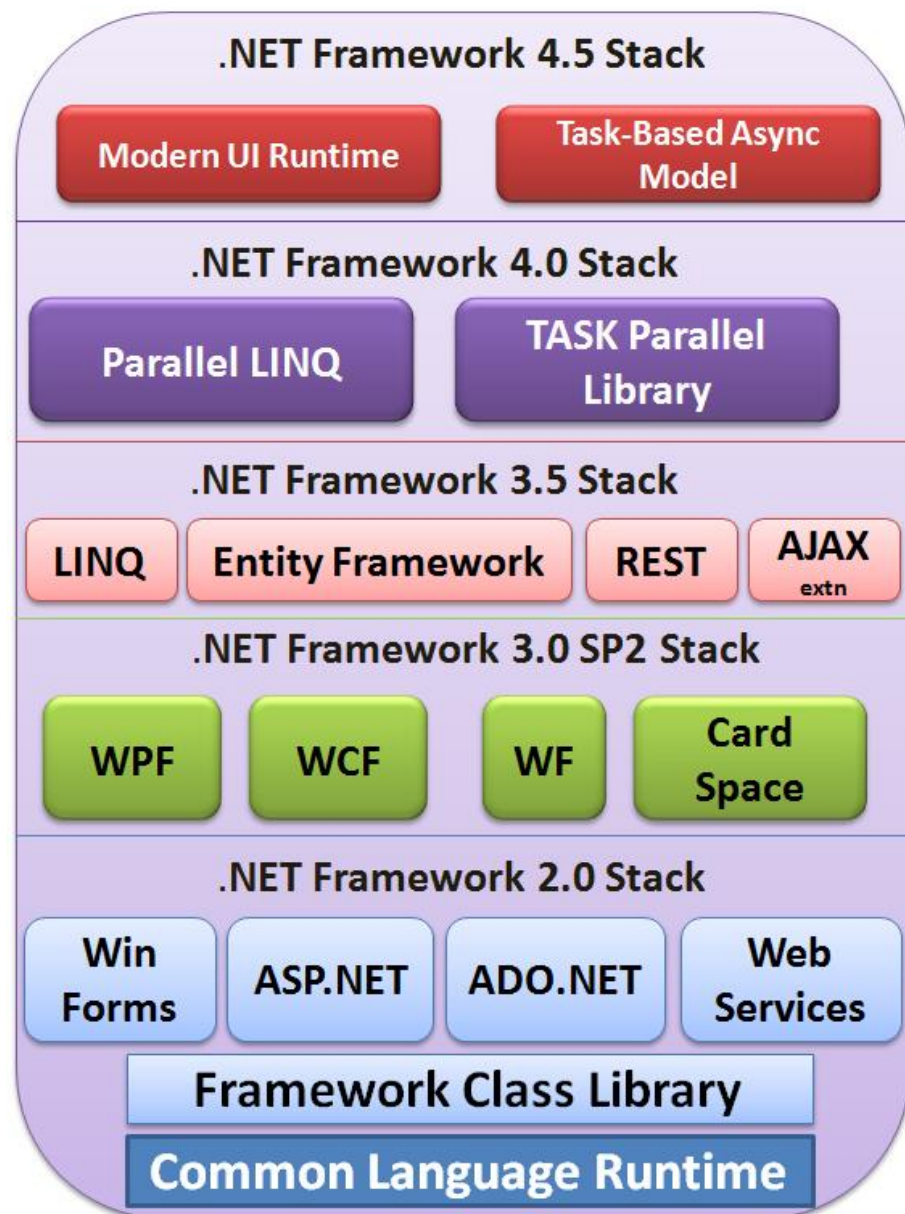
1.4. NỀN TẢNG CỦA .NET

.NET Framework là một phần cốt yếu của Windows để xây dựng và chạy các thể hệ tiếp theo của ứng dụng phần mềm và các dịch vụ Web XML.

.NET Framework được thiết kế để:

- Cung cấp 1 môi trường phù hợp cho lập trình hướng đối tượng
- Phát triển phần mềm đơn giản hơn và tránh xung đột về phiên bản bằng cách cung cấp một môi trường thực thi mã (code-execution environment)
- Thực hiện mã an toàn bằng cách cung cấp môi trường thực thi mã (code-execution environment)

1.5. CÁC THÀNH PHẦN .NET



Common Type System (CTS)

.NET framework hỗ trợ nhiều ngôn ngữ và đều dùng một thành phần gọi là hệ thống kiểu chung CTS trong CLR. CTS hỗ trợ một loạt kiểu và toán tử có thể thấy trong hầu hết các ngôn ngữ lập trình nên gọi một ngôn ngữ từ một ngôn ngữ khác sẽ không yêu cầu chuyển kiểu. Dẫn đến chúng ta có thể xây dựng các ứng dụng .NET sử dụng cả ngôn ngữ VB.NET lẫn C#, C++...

Common Language Specification (CLS)

Đặc tả ngôn ngữ chung CLS là một tập con của CTS, nó định nghĩa một tập các quy tắc cho phép liên kết hoạt động trên nền tảng .NET. Các quy tắc này sẽ trợ giúp và chỉ dẫn cho các nhà thiết kế compiler của hãng thứ 3 hoặc những người muốn xây dựng thư viện dùng chung.

1.6. USING .NET FRAMEWORK

Lập trình viên có thể phát triển ứng dụng bằng cách sử dụng một trong rất nhiều ngôn ngữ được .NET Framework hỗ trợ và sử dụng các lớp thư viện cơ sở. Ví dụ, để hiển thị một dòng thông báo lên màn hình, ta có thể dùng lệnh code sau:

```
System.Console.WriteLine(".NET Architecture");
```

Phương thức *WriteLn()* sẽ được dùng trên tất cả ngôn ngữ của .NET. Vì phương thức này được tạo nên bởi Framework Class Library như là một lớp chung cho tất cả các ngôn ngữ của .NET

1.7. CÁC THÀNH PHẦN .NET KHÁC

Base Framework Classes

Gồm nhiều lớp cung cấp các phương thức như là input/output, thao tác trên chuỗi, quản lý bảo mật, kết nối mạng, vv....vv

ASP.NET

Cung cấp một tập các lớp để xây dựng nên ứng dụng Web. Ứng dụng Web ASP.NET có thể được xây dựng bằng cách sử dụng Web Form, đó là một tập các lớp để thiết kế mẫu cho nhiều trang Web giống nhau như HTML. ASP.NET cũng hỗ trợ Web services, có thể truy cập bằng cách sử dụng các tiêu chuẩn của giao thức.

ADO.NET Cung cấp các lớp tương tác với cơ sở dữ liệu

WPF

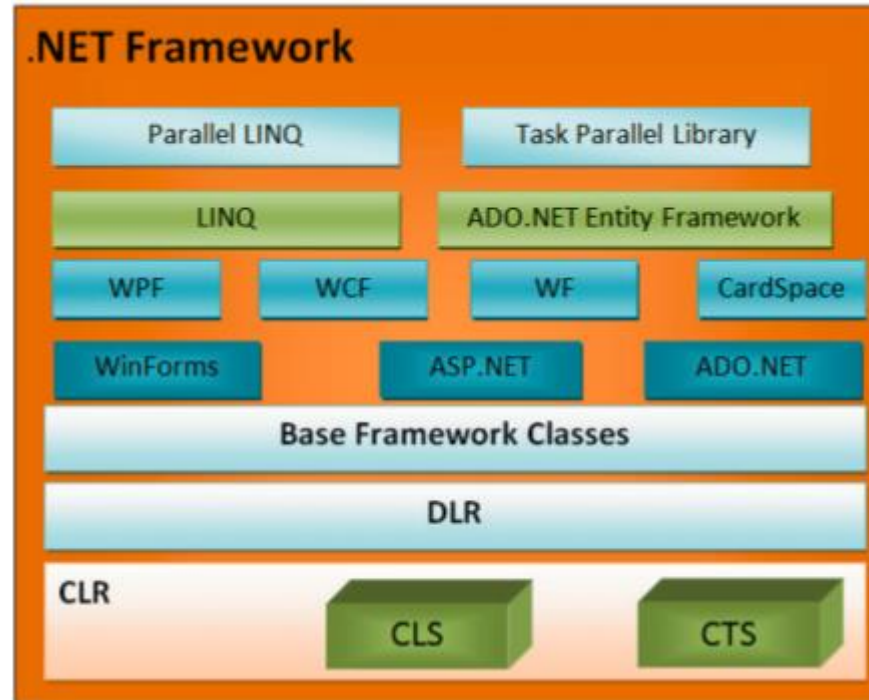
Là một nền tảng đồ họa dựa trên XML và vector. WPF sử dụng phần cứng đồ họa 3D của máy tính và công nghệ 3D để tạo nên các ứng dụng desktop với giao diện người dùng phong phú trên nền tảng Windows.

WCF

Là một dạng nền tảng dịch vụ định hướng thông tin. WCF cho phép tạo ra các endpoints và các chương trình có thể đồng bộ gửi và nhận dữ liệu từ endpoints

1.7. CÁC THÀNH PHẦN .NET KHÁC (2)

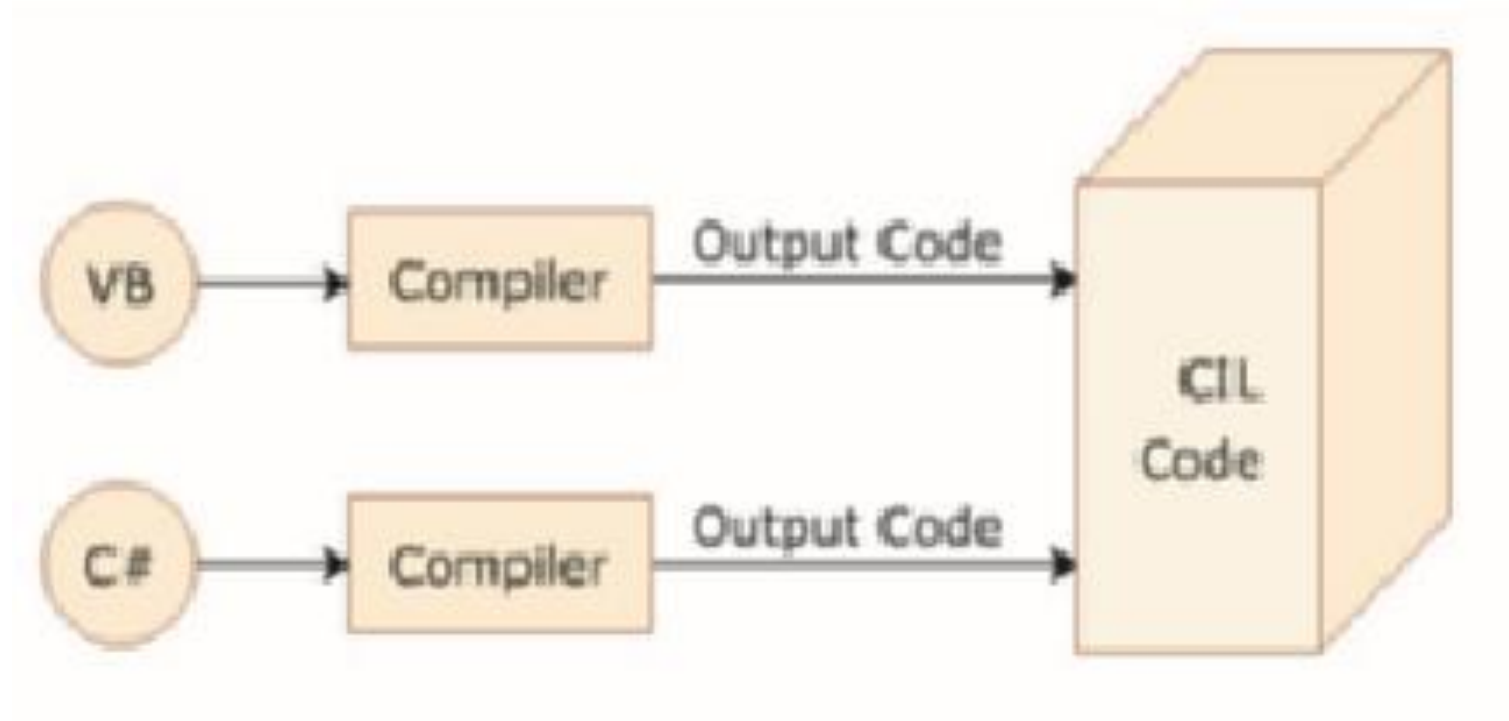
LINQ Là một thành phần cung cấp khả năng truy vấn dữ liệu cho một ứng dụng .NET
ADO.NET Entity Framework Gồm nhiều công nghệ xây dựng dựa trên ADO.NET cho phép tạo ra các ứng dụng trung tâm dữ liệu trong lập trình hướng đối tượng
Parallel LINQ Gồm nhiều lớp hỗ trợ lập trình song song bằng cách sử dụng LINQ
Task Parallel Library Là một thư viện đơn giản hóa việc lập trình song song trong ứng dụng .NET



1.8. COMMON INTERMEDIATE LANGUAGE (CIL)

- Mỗi ngôn ngữ lập trình .NET nói chung đều có một trình biên dịch và một môi trường run-time của riêng mình. Trình biên dịch chuyển mã nguồn thành mã thực thi có thể điều hành bởi người dùng.
- Một trong những mục tiêu chính của .NET Framework là kết hợp các môi trường run-time để các lập trình viên có thể làm việc với một tập các dịch vụ run-time.
- Khi code được viết bởi một ngôn ngữ .NET tương thích như là C# hay VB thì code sẽ được biên dịch, kết quả được xuất ra dưới dạng mã của MSIL. MSIL bao gồm một tập các cách thức mà mã được thực thi.

1.8. COMMON INTERMEDIATE LANGUAGE (CIL) (2)



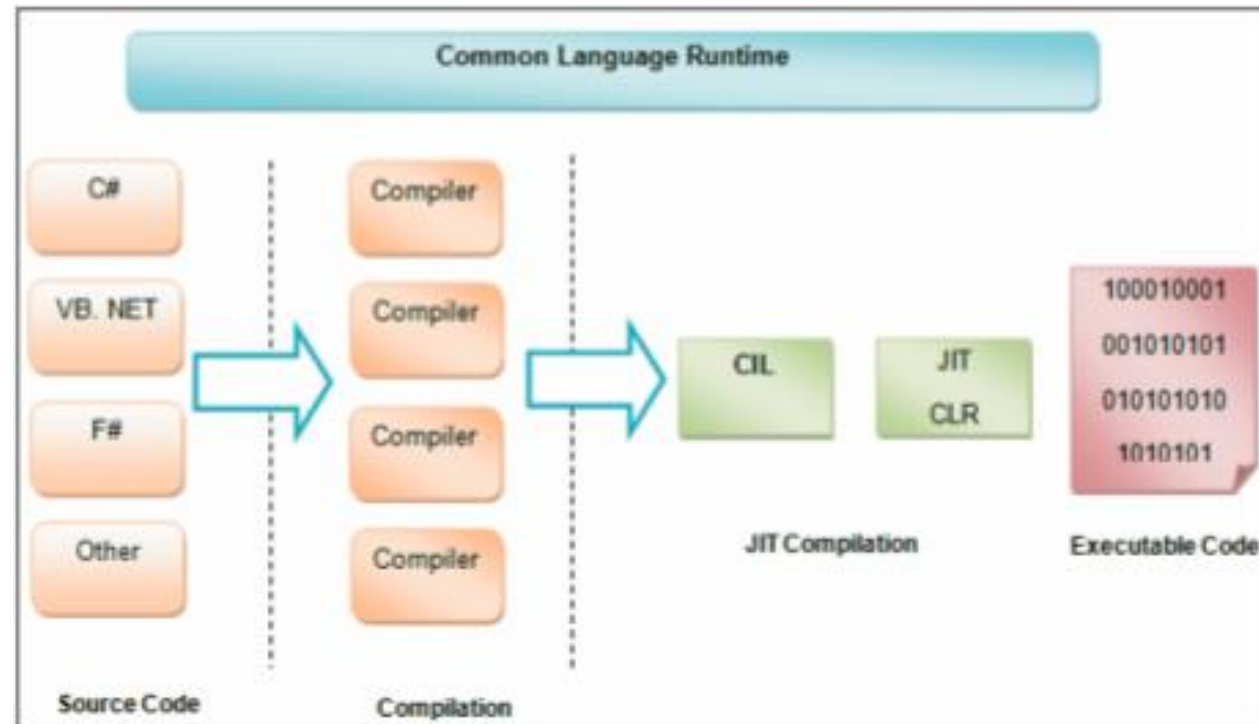
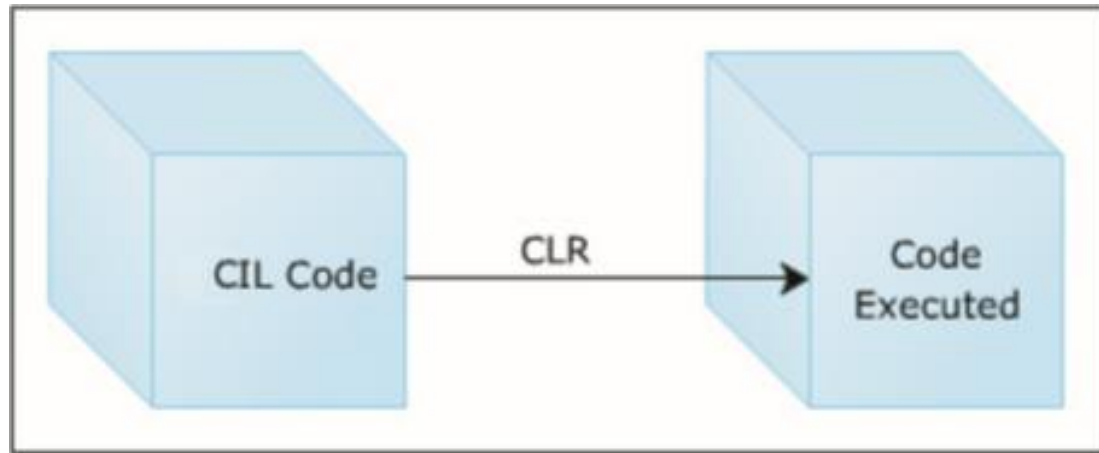
1.9. COMMON LANGUAGE RUNTIME (CLR)

CLR là nền tảng của .NET Framework. Quản lý thời gian chạy khi code đang thực thi các hoạt động như quản lý bộ nhớ, quản lý luồng, và làm việc từ xa. CLR hoạt động như engine để thực hiện mã của .NET Framework. Nó quản lý việc thực thi và cung cấp môi trường tương thích để thực thi cho các chương trình.

.NET Framework hỗ trợ một số công cụ phát triển và biên dịch ngôn ngữ trong Software Development Kit (SDK).

Do đó, CLR cung cấp một môi trường thực thi đa ngôn ngữ.

1.9. COMMON LANGUAGE RUNTIME (CLR) (2)



1.10. DYNAMIC LANGUAGE RUNTIME (DLR)

- DLR là một môi trường run-time xây dựng trên CLR cho phép khả năng tương tác giữa các ngôn ngữ động như Ruby và Python với .NET Framework.
- Các ngôn ngữ .NET Framework, như là C#, VB và J# là ngôn ngữ dạng tĩnh, nghĩa là các lập trình viên cần phải xác định kiểu của đối tượng khi phát triển chương trình.
- Mặt khác, trong ngôn ngữ động, lập trình viên không cần xác định kiểu của đối tượng trong giai đoạn phát triển.
- DLR cho phép tạo và chạy các ngôn ngữ động trên nền .NET Framework. Ngoài ra, DLR còn cung cấp các tính năng linh hoạt đối với các ngôn ngữ kiểu tĩnh hiện nay. Ví dụ, C# dựa trên DLR để thực hiện các ràng buộc động.

HỌC .NET NÊN CHỌN GÌ?

- Ngôn ngữ lập trình C#
- + Làm chủ ngôn ngữ lập trình.
- + Sử dụng thành thạo thư viện phổ biến .NET Framework
- Cơ sở dữ liệu:
 - + Hiểu rõ về lý thuyết của CSDL quan hệ.
 - + Thiết kế cơ sở dữ liệu trên Microsoft Sql Server
 - + Lập trình cơ sở dữ liệu với T-SQL trên Microsoft Sql Server.
- Phát triển ứng dụng DESKTOP
- Phát triển ứng dụng trên WEB
- Sử dụng thành thạo ADO.NET để truy cập CSDL từ môi trường lập trình.
- Sử dụng ADO.NET Entity Framework (ORM) và LINQ.
- Biết và vận dụng WebService, WCF để phát triển các ứng dụng hướng dịch vụ.
- Lập trình di động: Windows Phone

Học nâng cao để làm việc:

- Web (Front-End):
 - + Html5, JQuery, AngularJS, KnockoutJS, ...
- Database:
 - + Advanced SQL Server: Report Service, Analysis Service, Integration Service, Service Broker.
 - + SQL Azure (Database Cloud Computing)
- Programming: Design Pattern
- NoSql:
 - + MongoDB, RavenDb, CouchDB, Cassandra DB, ...
 - + Real-time: SignalR

2.1. TỔNG QUAN NGÔN NGỮ C#

- C# là một trong những ngôn ngữ lập trình trong họ .NET của Microsoft và khi lập trình C#, người lập trình phải cài đặt .NET Framework
- C# được phát triển bởi Anders Hejlsberg và team của ông trong quá trình phát triển .Net Framework
- C# được thiết kế cho Common Language Infrastructure(CLI), gồm Executable Code và Runtime Enviroment
- C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của 2 ngôn ngữ lập trình mạnh mẽ nhất là C++ và Java

2.2. TÍNH NĂNG CỦA C#

- Ngôn ngữ lập trình hướng đối tượng
- Tuân theo chuẩn ECMA
- Bộ thu gom rác tự động
- An toàn kiểu
- Hỗ trợ Generic
- Partial Classes
- Anonymous Methods (các phương thức bất động bộ, ẩn tên)
- Nullable Type
- Lập trình song song

2.3. CÁC ỨNG DỤNG CỦA C#

- **Trên windows:** C# với framework .NET được dùng để tạo ra các ứng dụng trên Windows như Microsoft Office, Visual Studio, Skype, Photoshop,...
- **Trên Web:** C# hỗ trợ lập trình viên tạo các ứng dụng web nhờ sự hỗ trợ của asp.net. Với ngôn ngữ này, các ứng dụng có thể chạy mượt mà trên máy chủ.
- **Trên Game: Unity**
- **Trên Mobile**
- **Trên điện toán đám mây: Azura**
- **Ứng dụng windows Store, Enterprise**
- **Thành phần, điều khiển:** C# còn được ứng dụng trong xây dựng nhiều thành phần của máy chủ. Đây là một trong các ứng dụng quan trọng của ngôn ngữ lập trình C#.

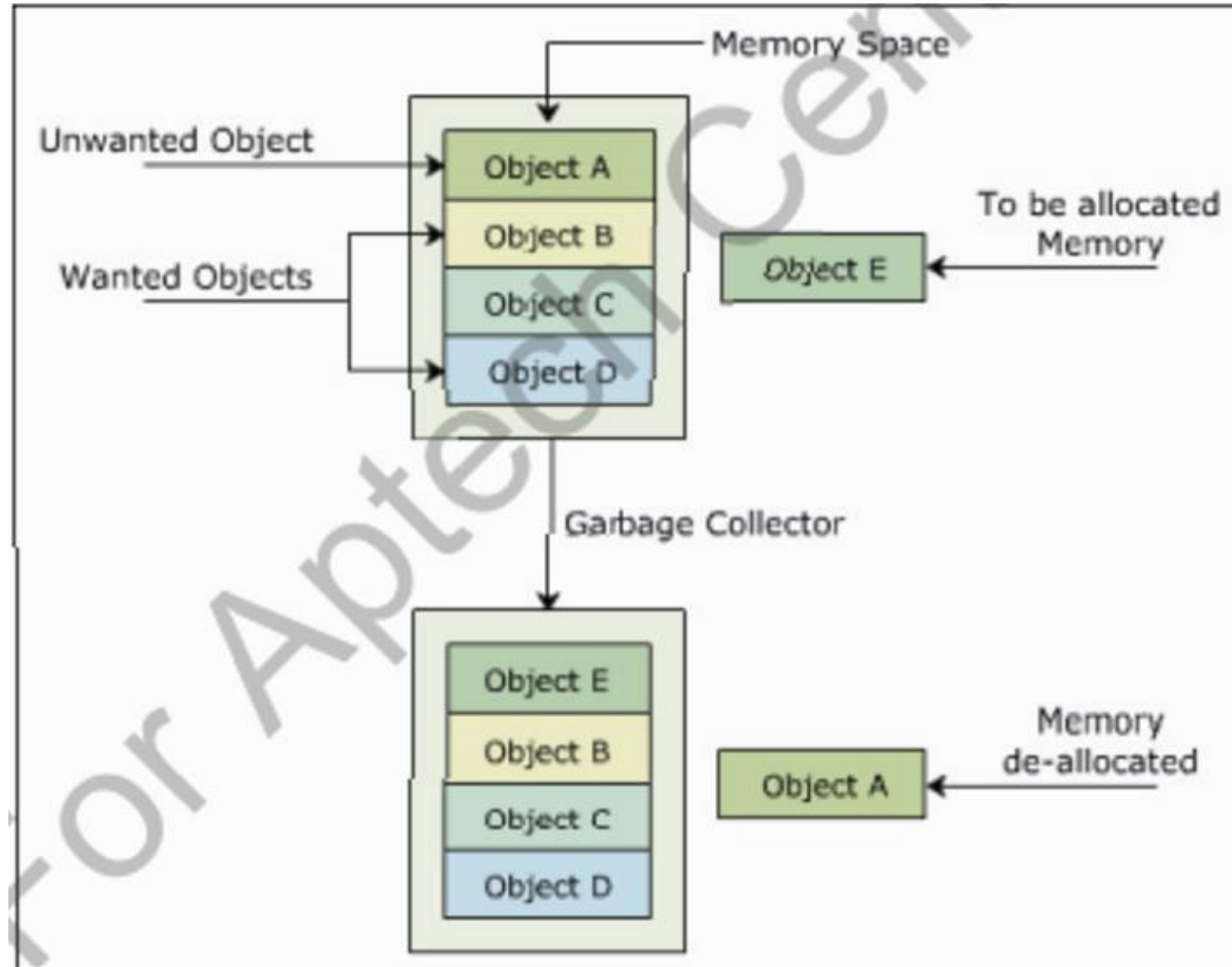
2.4. QUẢN LÝ BỘ NHỚ - MEMORY MANAMENT

Nếu đã từng lập trình C/C++ thì chắc hẳn sẽ nhớ đến một trong những điểm mạnh và cũng rất phức tạp của ngôn ngữ này là con trỏ và cấp phát động. Bạn luôn phải tuân thủ theo quy tắc: cấp bao nhiêu, thu hồi bấy nhiêu.

Trong .NET, khái niệm con trỏ bị hạn chế sử dụng (nếu muốn sử dụng thì phải đánh dấu đoạn mã là unsafe). Tuy nhiên, thực chất khi chúng ta tạo một đối tượng của lớp (nói rộng hơn là một kiểu tham chiếu) thì quá trình cấp phát bộ nhớ cũng tương tự như cấp phát động cho con trỏ trong C++

2.5. BỘ DỌN RÁC - GARBAGE COLLECTION

- Đây là một tiến trình đặc biệt có nhiệm vụ duyệt qua các vùng nhớ đã được cấp phát và kiểm tra xem vùng nhớ nào không còn được sử dụng nữa (không còn tham chiếu tới nó nữa) thì sẽ thực hiện thu hồi một cách tự động để có thể cấp phát cho các yêu cầu tiếp theo.
- Khi chương trình khởi chạy thì một vùng nhớ liên tục còn trống sẽ được dành riêng để cấp phát cho các biến trong chương trình (vùng nhớ này được gọi là managed-heap). Khi chúng ta dùng toán tử new để tạo một đối tượng mới thì chương trình sẽ kiểm tra xem vùng nhớ này còn đủ để cấp phát hay không, nếu không đủ thì GC sẽ được khởi động. Bước đầu tiên mà GC thực hiện là tạm dừng chương trình và thực hiện việc duyệt để đánh dấu tất cả những vùng nhớ đang được sử dụng với khởi đầu tại một điểm nào đó đã biết trước (hay được gọi là điểm gốc).



2.6. GIỚI THIỆU PHẦN MỀM VISUAL STUDIO

- Là phần mềm hỗ trợ đắc lực công việc lập trình. Thuộc quyền sở hữu của Microsoft. Năm 1997, phần mềm lập trình này có tên mã Project Boston. Sau đó, Microsoft đã kết hợp các công cụ phát triển, đóng gói thành sản phẩm duy nhất.
- Visual Studio là hệ thống tập hợp tất cả những gì liên quan tới phát triển ứng dụng, bao gồm trình chỉnh sửa mã, trình thiết kế, gỡ lỗi. Tức là, có thể viết code, sửa lỗi, chỉnh sửa thiết kế ứng dụng dễ dàng chỉ với 1 phần mềm mà thôi. Không dừng lại ở đó, người dùng còn có thể thiết kế giao diện, trải nghiệm trong Visual Studio như khi phát triển ứng dụng Xamarin, UWP bằng XAML hay Blend vậy.
- Các tính năng của Visual Studio: Đa nền tảng, đa ngôn ngữ lập trình, hỗ trợ website, kho tiện ích mở rộng phong phú, lưu trữ phân cấp, kho lưu trữ an toàn, màn hình đa nhiệm, hỗ trợ viết code, hỗ trợ thiết bị đầu cuối, hỗ trợ git, comment,...

2.7. CÁC PHIÊN BẢN CỦA VISUAL STUDIO

Visual studio Community

- Được xây dựng để phục vụ những nhà phát triển cá nhân, học viên lập trình và nghiên cứu học thuật.
- Bên cạnh đó công cụ này cũng có thể phục vụ các dự án nguồn mở, và các tổ chức phi doanh nghiệp có số lượng tối đa là 5 người.
- Luôn hỗ trợ nền tảng phát triển.
- Trong môi trường phát triển tích hợp: Nó hỗ trợ tích hợp một số tính năng sau đây: định nghĩa Peek, tái cấu trúc, triển khai web bằng một lần nhấp, trình xem tài nguyên mô hình và giải pháp trực quan hóa với đồ thị phụ thuộc cùng với bản đồ mã và đa mục tiêu.
- Hỗ trợ số liệu mã và gỡ lỗi đồ họa, tiến hành phân tích các mã tĩnh cùng với trung tâm chẩn đoán và hiệu suất với phần gỡ lỗi và chẩn đoán nâng cao hỗ trợ.
- Đây là phiên bản miễn phí nên chỉ cung cấp công cụ kiểm tra thử đơn vị cho người sử dụng để kiểm tra mã của mình.
- Đối với tích hợp phát triển đa nền tảng thì cung cấp các tính năng đa nền tảng có tên gọi Xamarin, gồm mã chia sẻ giữa các đối tác của ứng dụng iOS và Android, thiết kế giao diện người dùng gốc.
- Hỗ trợ phát triển và hợp tác với những nhà phát triển khác ngoài visual studio theo nhu cầu của người sử dụng.

2.7. CÁC PHIÊN BẢN CỦA VISUAL STUDIO (2)

Visual studio Professional

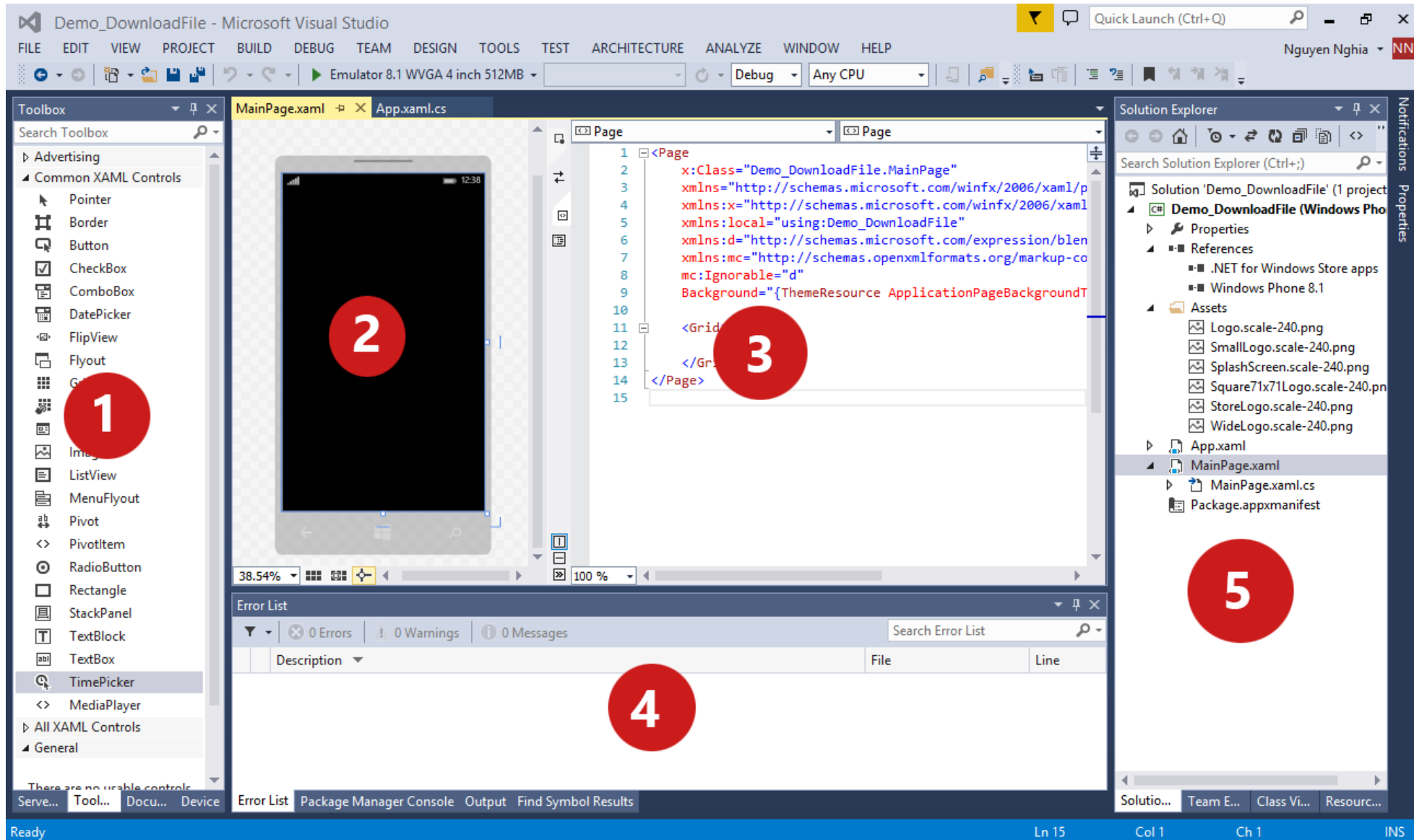
- Thường được sử dụng để phục vụ cho những doanh nghiệp là chủ yếu.
- Hỗ trợ nền tảng phát triển, hỗ trợ tích hợp mọi thứ mà phiên bản Community hỗ trợ, và thêm CodeLens.
- Bên cạnh đó phiên bản chuyên nghiệp này còn hỗ trợ gỡ lỗi và chẩn đoán và nâng cao.
- Phiên bản này chỉ cho phép hỗ trợ kiểm tra thử đơn vị cho người sử dụng để kiểm tra mã của mình.
- Cung cấp các tính năng phát triển đa nền tảng và cho phép người sử dụng kết hợp với các nhà phát triển khác.

2.7. CÁC PHIÊN BẢN CỦA VISUAL STUDIO (3)

Visual studio Enterprise

- Dành cho những nhà phát triển cá nhân, học viên, những người nghiên cứu học thuật, phục vụ các dự án mở và các tổ chức phi doanh nghiệp có số lượng không quá 5 người sử dụng và các doanh nghiệp.
- Hỗ trợ nền tảng phát triển và tích hợp mọi thứ mà phiên bản chuyên nghiệp hỗ trợ, gồm xác thực phụ thuộc trực tiếp, sơ đồ của lớp kiến trúc, xác thực kiến trúc cùng bản sao mã.
- Bên cạnh việc hỗ trợ tính năng gỡ lỗi và chẩn đoán nâng cao như bản chuyên nghiệp. Nó còn hỗ trợ tích hợp trình gỡ lỗi bản đồ mã và phân tích kết xuất bộ nhớ .NET đồng thời hỗ trợ cả IntelliTrace.
- Đối với các tính năng hỗ trợ công cụ kiểm tra: Giúp người dùng kiểm tra được nhiều mảng khác nhau và có tính năng kiểm tra thủ công.
- Cung cấp nhiều tính năng hơn hai phiên bản kia bởi nó bao gồm các hội đồng nhúng, trình kiểm tra, trình biên dịch Xamarin và trình mô phỏng.

2.8. MÔI TRƯỜNG IDE



2.9. TẠO PROJECT ĐẦU TIÊN

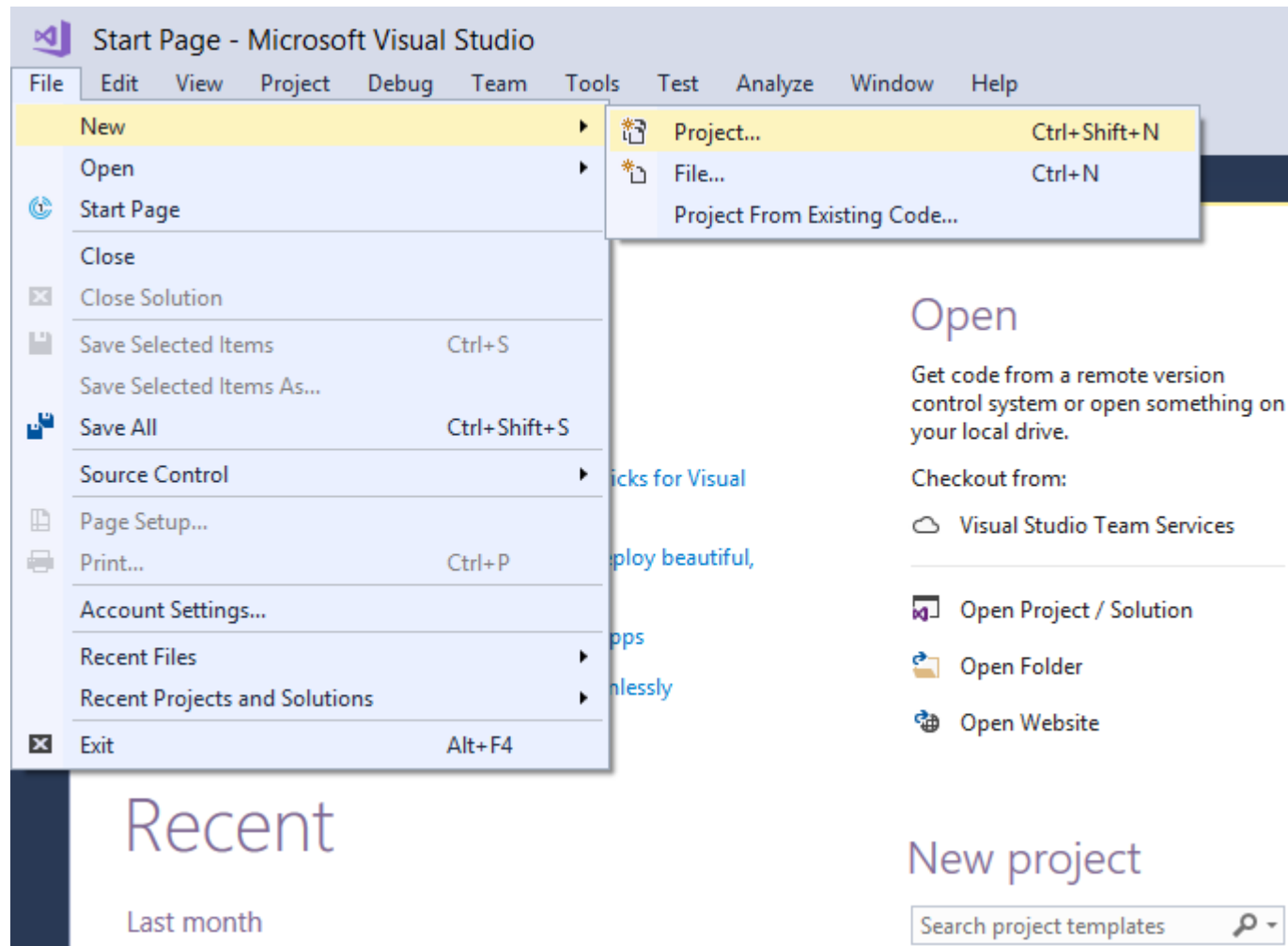
Ứng dụng Console, là ứng dụng không có giao diện.

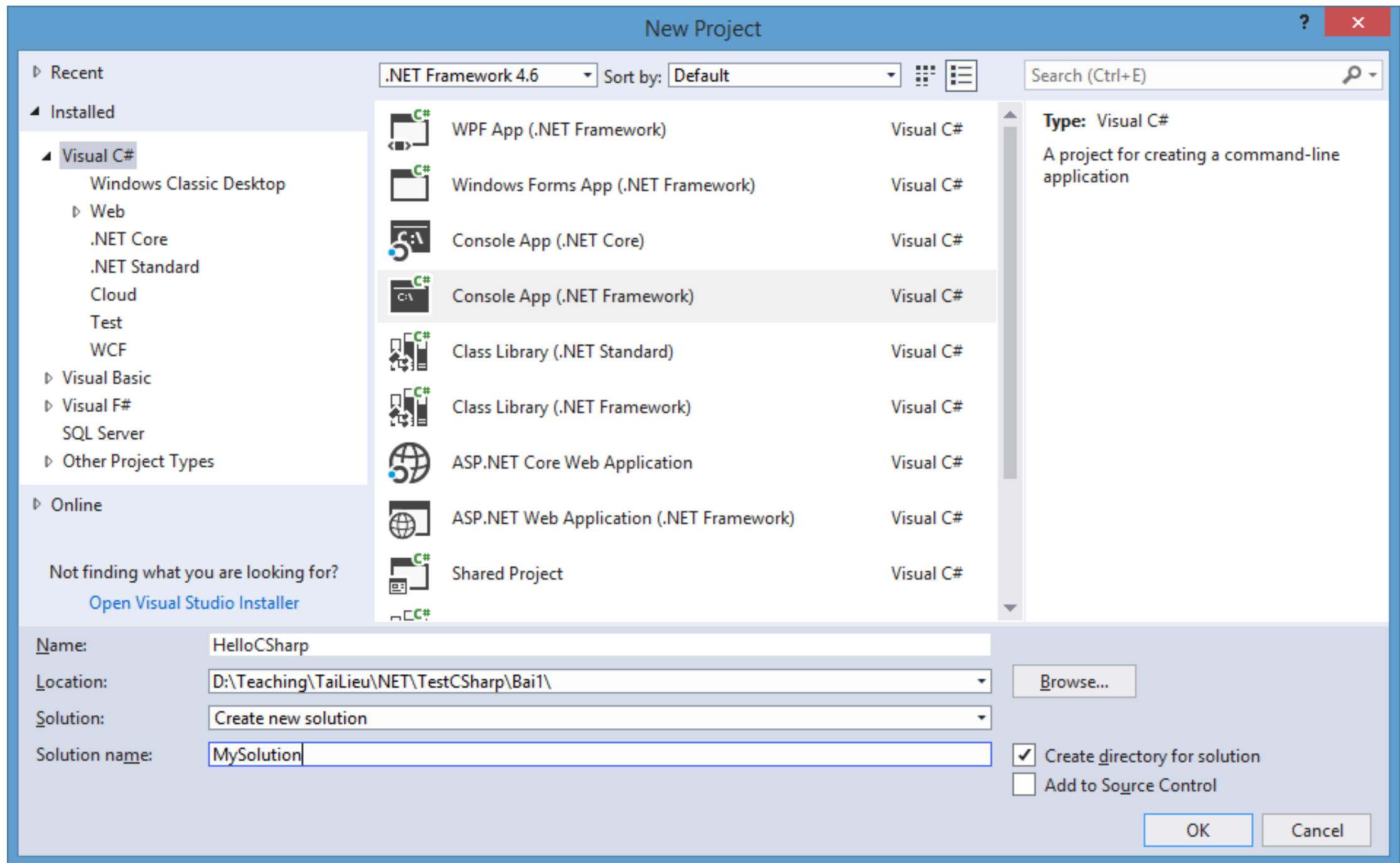
Nhập vào:

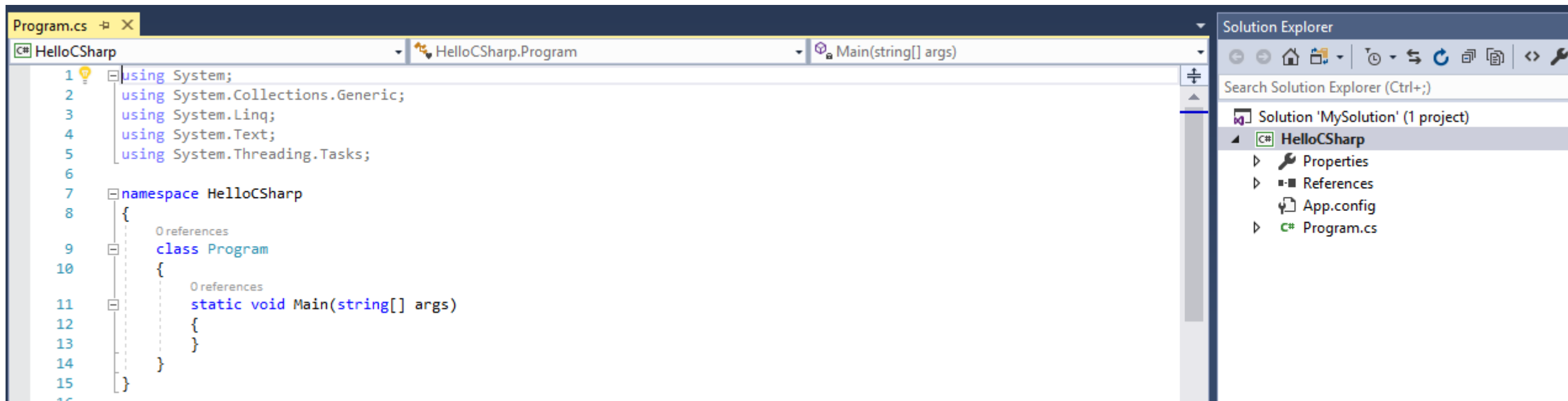
Name: HelloCSharp

Solution: Create new solution

Solution Name: MySolution







Program.cs

C# HelloCSharp HelloCSharp.Program

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace HelloCSharp
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("Hello CSharp");
14             Console.ReadLine();
15         }
16     }
17 }
```

D:\Teaching\Tailieu\NET\TestCSharp\Bai1\MySolution\HelloCSharp\bin\Debu...

```
Hello CSharp
```