

# .NET Framework và ngôn ngữ lập trình Visual C#

Hướng dẫn tự học lập trình C# toàn tập > .NET Framework và ngôn ngữ lập trình Visual C#

C# là một trong những ngôn ngữ hoạt động trên một nền tảng đặc biệt gọi là “.NET Framework”. Chương trình viết bằng C# (và các ngôn ngữ .NET khác) có quy trình dịch và cách thực thi khác biệt với các ứng dụng windows “bình thường”. Sự khác biệt này có ảnh hưởng lớn đến cách nghĩ và cách code trong C#.

Do vậy, các tài liệu về lập trình C# hầu như luôn phải giới thiệu về .NET Framework trước khi bắt đầu nói về chính ngôn ngữ.

Bài học này sẽ giúp bạn hiểu .NET Framework là gì, sự khác biệt về quy trình biên dịch – thực thi ứng dụng trong .NET framework với các loại ứng dụng truyền thống. Cuối cùng bạn sẽ làm quen với một số đặc điểm của ngôn ngữ lập trình C#.

## NỘI DUNG CỦA BÀI [ Ấn ]

1. .NET framework là gì?
  - 1.1. .NET framework là một bộ thư viện lớp
  - 1.2. .NET framework là một môi trường thực thi cho ứng dụng
  - 1.3. Một số công nghệ trong .NET Framework
  - 1.4. Cài đặt .NET framework trên windows
2. Biên dịch mã nguồn và thực thi chương trình trong .NET framework
  - 2.1. Biên dịch và thực thi chương trình native
  - 2.2. Biên dịch và thực thi chương trình trong .NET framework
3. Ngôn ngữ lập trình C#
  - 3.1. Một số đặc điểm của ngôn ngữ C#
  - 3.2. Trình biên dịch C#
  - 3.3. Một số điểm cần lưu ý
4. Kết luận

## .NET framework là gì?

Với câu hỏi “.NET Framework là gì” bạn sẽ nhận được hai câu trả lời: (1) .NET Framework là một bộ thư viện lớp; (2) .NET Framework là một môi trường thực thi của ứng dụng.

### .NET framework là một bộ thư viện lớp

.NET framework chứa một bộ thư viện rất lớn chứa các loại công cụ khác nhau giúp người lập trình xử lý được hầu hết các công việc thường gặp trong quá trình phát triển ứng dụng. Vì vậy, lập trình ứng dụng với .NET framework thường có hiệu quả rất cao và thời gian làm việc ngắn hơn.

Người lập trình có thể sử dụng bất kỳ ngôn ngữ nào mà trình biên dịch của nó có thể dịch mã nguồn sang một dạng mã trung gian theo yêu cầu của .NET framework. Visual C#, Visual Basic.NET, Visual C++ .NET, và F# là bốn ngôn ngữ chính thức được Microsoft phát triển để hoạt động trên .NET framework.

Ngoài ra còn một số ngôn ngữ khác không do Microsoft phát triển cũng được thiết kế hướng tới nền tảng .NET như Delphi.NET, Oxygence (hai ngôn ngữ có cấu trúc cú pháp tương tự Pascal), IronPython (phiên bản của Python dịch sang .NET framework), v.v..

Các ngôn ngữ hướng tới .NET framework ngoài việc có thể truy cập tới bộ thư viện còn có thể sử dụng thư viện người dùng viết bằng các ngôn ngữ .NET khác. Ví dụ, người phát triển ứng dụng trên C# hoàn toàn có thể sử dụng thư viện do người khác phát triển trên Visual Basic.

## .NET framework là một môi trường thực thi cho ứng dụng

Chương trình viết cho .NET framework không thực thi trực tiếp trong môi trường hệ điều hành mà thực thi trong khuôn khổ của chương trình CLR và được quản lý bởi chương trình này.

Vì thực thi trong một môi trường riêng khép kín và được quản lý chặt chẽ, chương trình .NET có thể tránh được nhiều lỗi thường gặp trong quá trình phát triển ứng dụng.

Do môi trường thực thi của các ứng dụng .NET cung cấp nhiều tính năng cao cấp (như quản lý bộ lý, xử lý ngoại lệ, v.v.), việc lập trình ứng dụng trên .NET framework đơn giản hơn.

Việc học lập trình với một ngôn ngữ bất kỳ của .NET framework có giá trị rất lớn khi ta có thể sử dụng nó để viết hầu như bất kỳ loại ứng dụng nào, có thể viết ứng dụng không chỉ ứng dụng chạy trên Windows mà còn có thể cho các nền tảng khác.

## Một số công nghệ trong .NET Framework

Dưới đây là một số [công nghệ phổ biến trong .NET framework](#)

- Console: xây dựng ứng dụng với giao diện dòng lệnh;
- [Windows Forms](#): xây dựng ứng dụng desktop (giao diện đồ họa) cho windows;
- Windows Presentation Foundation: công nghệ mới xây dựng ứng dụng desktop (giao diện đồ họa) cho windows;
- ASP.NET: nền tảng để phát triển các ứng dụng web chạy trong chương trình máy chủ IIS, bao gồm ASP.NET Web Forms, ASP.NET MVC, ASP.NET Web API;
- [ADO.NET và Entity Framework](#): công nghệ cho phép chương trình kết nối và sử dụng cơ sở dữ liệu quan hệ (SQL Server, MySQL, v.v.);
- Windows Communications Foundation: công nghệ cho phép phát triển ứng dụng mạng hướng dịch vụ (Service Oriented Application, SOA).

Hiện nay bạn có thể sẽ nghe thấy cả [.NET Core](#), hay [.NET 5](#) (sắp ra mắt). Lưu ý rằng, mặc dù đều có chung phần ".NET" nhưng chúng là những nền tảng khác nhau. Để phân biệt, người ta thường nói rõ ".NET Framework" hay ".NET Core". Trong đó, .NET Framework là nền tảng thuần túy cho windows, .NET Core hoạt động đa nền tảng (trong đó có cả windows), .NET 5.0 là nền tảng hợp nhất trong tương lai.

## Cài đặt .NET framework trên windows

Trên hệ điều hành Windows, .NET framework có thể cài đặt trên phiên bản Windows 98 và mới hơn, hoặc Windows NT 4.0 về sau. Các phiên bản cũ hơn của Windows không cho phép cài đặt .NET framework. Các hệ điều hành mới như Windows 10 đều mặc định cài đặt .NET và có thể tự động cập nhật phiên bản mới.

Các chương trình viết cho .NET framework chỉ có thể hoạt động nếu trên hệ điều hành có cài đặt phiên bản tương đương hoặc mới hơn của .NET (so với phiên bản sử dụng cho quá trình phát triển ứng dụng).

Phiên bản mới nhất của .NET Framework hiện nay là 4.7. Nếu bạn đang dùng Windows 10, hoặc bạn đã [cài đặt Visual Studio](#), máy tính của bạn đã có sẵn .NET bản mới nhất.

## Biên dịch mã nguồn và thực thi chương trình trong .NET framework

.NET framework và các ngôn ngữ hỗ trợ nền tảng này hoạt động không giống như các chương trình “bình thường” khác trong Windows.

### Biên dịch và thực thi chương trình native

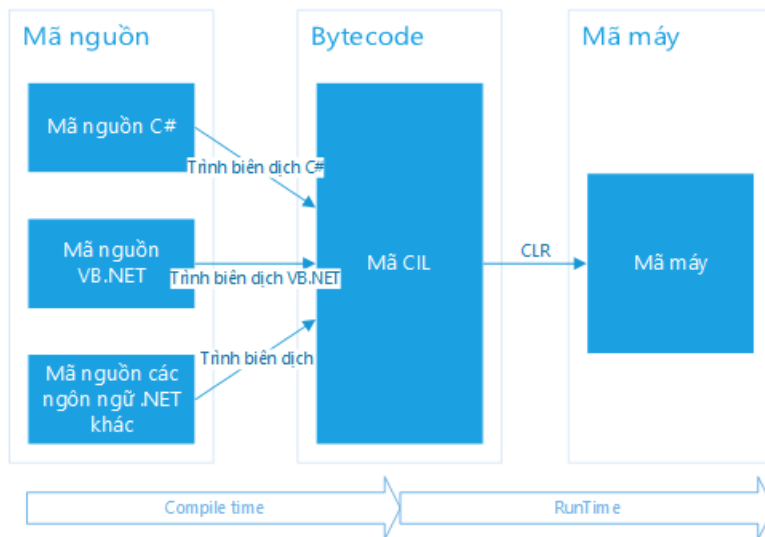
Các file mã nguồn của chương trình viết bằng một ngôn ngữ lập trình nào đó sẽ được chương trình dịch của ngôn ngữ đó chuyển thành file chương trình chứa mã máy (native code/instruction). Giai đoạn này gọi là **Compile time**.

Khi người dùng chạy ứng dụng, file chương trình được hệ điều hành tải vào bộ nhớ và bắt đầu thực hiện các lệnh chứa trong đó. Hệ điều hành đóng vai trò môi trường hoạt động và là người quản lý việc thực thi của ứng dụng. Giai đoạn tải và thực thi các lệnh gọi là **Runtime**.

Loại chương trình được hệ điều hành tải, thực thi và quản lý trực tiếp như vậy thường được gọi là **ứng dụng native**. Ứng dụng native trong windows được tạo ra với các ngôn ngữ như C/C++, Pascal, Delphi, Visual basic, v.v..

### Biên dịch và thực thi chương trình trong .NET framework

Trong .NET framework, mã nguồn viết bằng các ngôn ngữ lập trình không được biên dịch trực tiếp thành mã máy để thực thi (trực tiếp trong hệ điều hành) như các chương trình viết bằng C/C++ hay Pascal/Delphi.



Quy trình biên dịch và thực thi chương trình trong .NET framework

Mã nguồn viết bằng một ngôn ngữ .NET (C#, Visual basic .NET) được trình biên dịch của ngôn ngữ đó dịch thành tệp mã đặc biệt gọi là tệp *mã CIL* (Common Intermediate Language). CIL là một loại ngôn ngữ trung gian đặc biệt được Microsoft tạo ra cho nền tảng .NET.

CIL, trước đây còn được gọi là MSIL (Microsoft Intermediate Language), cũng chính là một ngôn ngữ lập trình. Đây là ngôn ngữ cấp thấp nhất mà người lập trình còn có thể đọc được code. Mã CIL cũng thường được gọi là bytecode.

Ở giai đoạn Runtime, một chương trình dịch đặc biệt có tên gọi là *JIT* (Just-in-time compiler) đọc và dịch tiếp mã CIL thành mã máy để thực thi. Quá trình thực thi này được kiểm soát bởi một chương trình đặc biệt gọi là *CLR* (Common Language Runtime).

Như vậy có thể hình dung (gần đúng) rằng, chương trình viết bằng một ngôn ngữ .NET sẽ chạy bên trong một chương trình khác (CLR) và bị chương trình CLR này kiểm soát. Cả CIL và CLR đều là các thành phần của một hệ thống tổng thể gọi là Common Language Intermediate (CLI).

Do việc biên dịch tất cả các ngôn ngữ .NET đều tạo ra CIL nên, ví dụ, một object tạo ra bằng Visual Basic .NET có thể được truy xuất từ code viết trên C#.

## Ngôn ngữ lập trình C#

C# là một ngôn ngữ lập trình được phát triển riêng biệt cho .NET framework với phiên bản C# 1.0 ra đời vào năm 2002 cùng với .NET framework 1.0. Phiên bản hiện tại của C# (ở thời điểm viết tài liệu này) là 7.

### Một số đặc điểm của ngôn ngữ C#

Ngôn ngữ C# chịu ảnh hưởng của nhiều ngôn ngữ lập trình khác, trong đó có C++, Eiffel, Java. Lập trình viên đã làm việc với các ngôn ngữ như C/C++ hay Java sẽ rất dễ dàng tiếp

thu C#.

C# được thiết kế theo hướng đơn giản, hiện đại, đa chức năng và hỗ trợ hầu hết các nguyên tắc lập trình hiện có:

- định kiểu mạnh (strong typing),
- hướng mệnh lệnh (imperative programming),
- hướng khai báo (declarative programming),
- hướng hàm (functional programming),
- hướng đối tượng (object-oriented programming),
- hướng thành phần (component-oriented programming),
- lập trình tổng quát (generic programming).

Bạn không nhất thiết phải hiểu hết các thuật ngữ “kỳ lạ” trên. Ở đây chỉ liệt kê ra nhằm chứng minh sự mạnh mẽ và phong phú của ngôn ngữ C#. Tuy nhiên, một số nguyên tắc trong số đó có ảnh hưởng rất lớn và trực tiếp đến việc học lập trình C#.

## Trình biên dịch C#

Để dịch mã nguồn C# thành chương trình cần sử dụng một trình biên dịch (compiler) cho ngôn ngữ này.

Hiện nay đang tồn tại một số trình biên dịch C# khác nhau:

- Microsoft Visual C# của Microsoft được xem là chương trình “chính thống”;
- Mono và trình biên dịch C# mã nguồn mở (tương đương với .NET 3.5, đồng thời cũng không hỗ trợ toàn bộ các class của .NET framework);
- DotGNU và trình biên dịch C# mã mở (tương đương với .NET 2.0);
- Trình biên dịch C# của Microsoft’s Rotor project (chỉ hỗ trợ tới C# 2.0 Windows XP).

Nếu bạn đã cài đặt sẵn Visual Studio, bạn không cần để ý đến trình biên dịch C# nữa. Mọi thứ đã setup sẵn sàng!

## Một số điểm cần lưu ý

C# có cú pháp, các cấu trúc điều khiển, một số kiểu dữ liệu cơ sở, v.v. rất giống C/C++ và Java. Nếu bạn có xuất phát điểm từ các ngôn ngữ này có thể dễ dàng nắm được các thành phần cơ bản của C#. Tuy nhiên đừng để sự tương đồng này đánh lừa bạn.

C# là ngôn ngữ hướng đối tượng 100%. Tức là mọi thứ trong C# đều là class. Do đó bạn sẽ không bao giờ nghe thấy các khái niệm như hàm toàn cục, biến toàn cục trong C#. Đặc điểm này giống với Java. Nếu xuất phát từ C/C++ bạn nên lưu ý vấn đề này.

Khi học lập trình C#, bạn đồng thời phải học thư viện .NET, mà trước hết là các thành phần cơ bản của thư viện này. Sau đó bạn phải đầu tư thời gian cho các kỹ thuật nâng cao của

.NET và C#. Cuối cùng bạn nên lựa chọn và đi sâu vào các công nghệ xây dựng trên nền tảng .NET.

Do đó, đừng suy nghĩ kiểu đốt cháy giai đoạn, muốn nhảy ngay vào các công nghệ (như [winform](#), [asp.net](#)). Nắm càng chắc C# và .NET, bạn càng dễ dàng tiếp cận các công nghệ. Nếu nhảy ngay vào công nghệ, bạn chỉ học được cái vỏ chứ không thể tiến xa được.

## Kết luận

---

Bài học này đã trình bày sơ lược về .NET Framework, các đặc điểm của nền tảng này, cũng như giới thiệu về ngôn ngữ lập trình C#.

Cũng lưu ý rằng, bài học này chỉ nêu ra những vấn đề rất cơ bản, vắn tắt, sơ lược về .NET và C# để giúp bạn nắm được tư tưởng và khái niệm chung. Còn rất nhiều nội dung mà trong khuôn khổ một bài học ngắn không thể trình bày được.

- + Nếu bạn thấy site hữu ích, trước khi rời đi hãy **giúp đỡ** site bằng một hành động nhỏ để site có thể phát triển và phục vụ bạn tốt hơn.
  - + Nếu bạn thấy bài viết hữu ích, hãy giúp **chia sẻ** tới mọi người.
  - + Nếu có thắc mắc hoặc cần trao đổi thêm, mời bạn viết trong phần **thảo luận** cuối trang.
- Cảm ơn bạn!