

Cấu trúc dự án C# – solution, project, file mã nguồn

Hướng dẫn tự học lập trình C# toàn tập > Cấu trúc dự án C# – solution, project, file mã nguồn

Solution và project là hai cấp độ quản lý file mã nguồn của C# và các thành phần hỗ trợ khác. Trong mỗi file mã nguồn, namespace là cấp độ quản lý code cao nhất. Tất cả những khái niệm này và cách làm việc với chúng đóng vai trò xương sống để có thể làm việc với C#.

Bài học này sẽ giúp bạn hiểu chi tiết các thành phần nêu trên.

NỘI DUNG CỦA BÀI [Ấn]

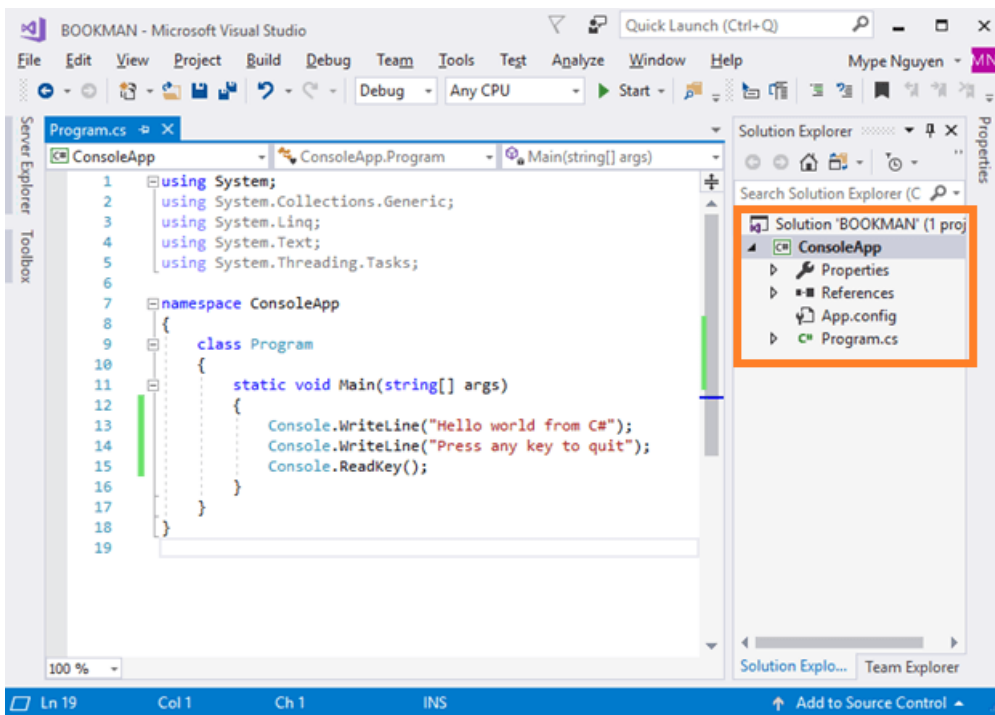
1. Cấu trúc solution/project của C#
 - 1.1. Project trong C# là gì?
 - 1.2. Solution trong C# là gì?
 - 1.3. Quy ước đặt tên solution và project của C#
2. Cấu trúc file/thư mục của C# project
 - 2.1. Thư mục bin
 - 2.2. .NET assembly
3. Kết luận

Cấu trúc solution/project của C#

C# quản lý mã nguồn theo cấu trúc cây gần giống với cấu trúc thư mục và bao gồm hai cấp độ cơ bản: *Project* và *Solution*.

Trong nội dung thực hành của bài [cài đặt Visual Studio](#), chúng ta đã tạo ra một project thuộc loại Console App bên trong một solution. Theo đó *solution* được đặt tên là BOOKMAN, còn *project* đặt tên là ConsoleApp.

Trong suốt giai đoạn học lập trình C# cơ bản chúng ta sẽ chỉ xây dựng ứng dụng với giao diện dòng lệnh (Console App). Đây là loại ứng dụng đơn giản nhất thường sử dụng trong việc học lập trình cơ bản.



Solution và Project trong C#

Project trong C# là gì?

Project (dự án) là cấp độ quản lý mã nguồn quan trọng nhất của C#. Mỗi project sau khi biên dịch sẽ tạo ra một chương trình.

Mỗi project mặc định đều chứa:

- các *file mã nguồn*: là các file văn bản có phần mở rộng .cs (viết tắt của C sharp);
- các *file cấu hình* của chương trình: là file xml có phần mở rộng .config;
- các *thư viện được tham chiếu* tới (References): là danh sách các file thư viện chuẩn của .NET framework, hoặc thư viện từ các hãng thứ ba, hoặc chính các project khác, chứa các class được sử dụng bởi các class trong project này.
- các *thuộc tính* (Properties): bao gồm nhiều loại thông tin khác nhau quyết định những tính chất quan trọng của project, như phiên bản của .NET framework được sử dụng, loại chương trình mà dự án này sẽ được dịch thành, các tài nguyên được sử dụng trong project, cấu hình của ứng dụng, v.v.. Visual Studio cung cấp giao diện đồ họa để có thể dễ dàng quản lý các thông tin này. Giao diện này mở ra khi click đúp vào mục Properties của project.

Tất cả các thành phần của một project đều đặt chung trong một thư mục cùng tên với project.

Solution trong C# là gì?

Solution (giải pháp) là cấp độ quản lý mã nguồn cao nhất trong C# cho phép quản lý tập trung nhiều project.

Mỗi solution trong C# có thể chứa nhiều project. Nếu solution không chứa project nào, nó gọi là Empty Solution. Tại mỗi thời điểm Visual Studio chỉ có thể mở một solution.

Trên giao diện Visual Studio, solution và các project của nó được hiển thị trong một cửa sổ riêng gọi là Solution Explorer. Cửa sổ này hiển thị tất cả các thành phần trong dự án C# theo cấu trúc cây, với solution làm gốc, các project là các nhánh trực tiếp xuất phát từ gốc này.

Ngoài ra, để tiện lợi trong việc quản lý các project thành viên, solution cho phép tạo thêm các thư mục, gọi là Solution Folder, trong đó lại có thể chứa các project khác. Cấu trúc quản lý này cho phép quản lý một số lượng lớn project một cách dễ dàng. Các project trong cùng một solution thường có quan hệ nhất định với nhau.

Nếu vô tình đóng cửa sổ Solution Explorer có thể mở lại bằng cách chọn View => Solution Explorer hoặc bấm tổ hợp Ctrl + Alt + L.

Quy ước đặt tên solution và project của C#

Đối với solution, quy tắc đặt tên tương tự như đối với file và thư mục trong windows.

Tuy nhiên, theo “truyền thống” của C#, tên solution nên đặt theo quy ước: (1) bắt đầu bằng chữ hoa, (2) sử dụng PascalCasing (viết hoa chữ đầu mỗi từ) nếu chứa nhiều từ, (3) không nên chứa dấu cách hoặc số.

Đối với project, quy tắc đặt tên chặt chẽ hơn vì tên project mặc định sẽ được sử dụng làm namespace (không gian tên – xem trong phần tiếp theo) của tất cả các file mã nguồn của project đó. Tuy nhiên, Visual Studio có thể tự động chuyển đổi tên cho phù hợp quy tắc.

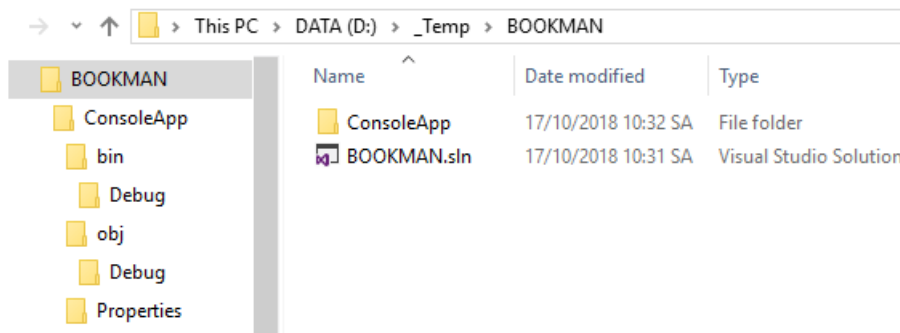
Một trong những quy ước thường được sử dụng để đặt tên project là: (1) chứa tên solution; (2) đặt tên riêng của project và phân tách với phần solution bằng một dấu chấm; (3) tuân thủ các quy ước đặt tên solution ở trên.

Quy ước đặt tên solution và project này rất có ích vì nó giúp tự động tạo ra cấu trúc namespace (xem dưới đây) phù hợp cho các file mã nguồn.

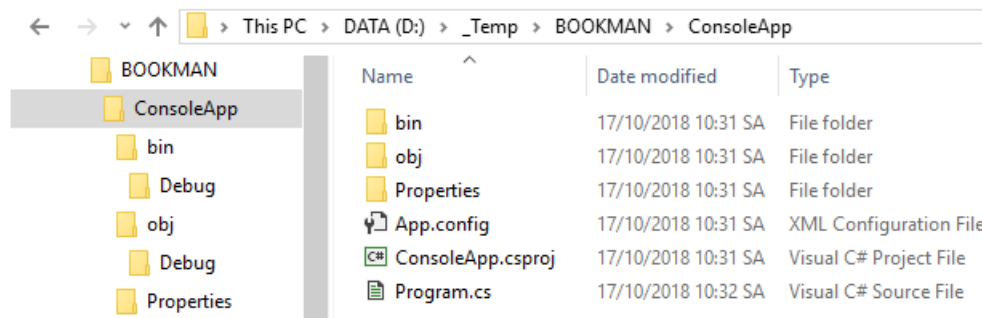
Cấu trúc file/thư mục của C# project

Nếu ở giai đoạn tạo project chúng ta đánh dấu lựa chọn “Create directory for solution”, Visual Studio sẽ tạo ra một thư mục có tên được xác định trong mục “Solution name”. Mỗi project được tạo ra sẽ đặt trong một thư mục con của thư mục solution và có cùng tên với project. Tất cả file của một project sẽ nằm trong thư mục này.

Toàn bộ cấu hình của solution được lưu trong một file có phần mở rộng **.sln** nằm trong thư mục chứa solution. Thông tin cấu hình của dự án được lưu trong file có tên trùng tên dự án và phần mở rộng **.csproj**.



Thư mục solution



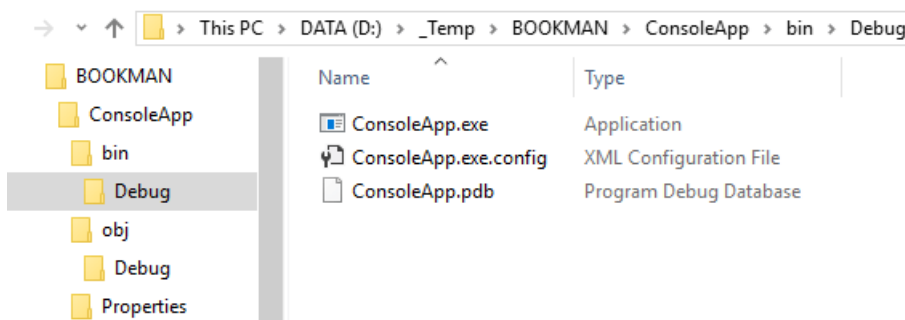
Thư mục project

Thư mục bin

Sau khi biên dịch project thành công, trong thư mục của nó sẽ xuất hiện một thư mục con có tên là *bin*. Thư mục này không xuất hiện trong cấu trúc dự án hiển thị ở Solution Explorer.

Nếu biên dịch ở chế độ debug, trong thư mục bin sẽ xuất hiện thư mục con Debug. File chương trình sau khi biên dịch (ở chế độ debug) xong sẽ xuất hiện trong thư mục này.

Nếu chạy chương trình ở chế độ debug (phím F5), thư mục Debug này sẽ trở thành thư mục hiện hành của chương trình đang chạy thử. Tất cả các file cấu hình và file tài nguyên (nếu có) cũng sẽ được tự động copy vào thư mục này.



Thư mục bin\Debug

Như vậy, nếu bạn muốn tìm file chạy của chương trình sau khi biên dịch, hãy mở theo đường dẫn **{tên solution}\{tên project}\bin\{Debug}**.

.NET assembly

Chúng ta lúc trước có nói rằng mỗi project sau khi biên dịch xong sẽ thành một chương trình. Tuy nhiên cách nói “chương trình” không hoàn toàn phù hợp đối với .NET framework.

Trong bài học về [.NET framework](#) chúng ta đã biết, mỗi project sau khi biên dịch đều trở thành một file chứa bytecode CIL. File mã CIL này được gọi là *Assembly*. .NET framework phân biệt hai loại assembly: một loại có thể tự nạp vào CLI và thực thi; một loại không thể tự mình nạp vào CLI mà cần phải có một assembly thuộc loại thứ nhất gọi, hoặc được một tiến trình khác gọi.

Loại assembly thứ nhất được lưu trong file có phần mở rộng .exe, tương tự như các file chương trình thực thi khác trong Windows. Loại assembly thứ hai được lưu trong các file có phần mở rộng .dll (Dynamic Link Library), tương tự như các file thư viện của Windows. Việc biên dịch ra .exe hay .dll phụ thuộc vào loại project. Ứng dụng console chúng ta vừa tạo thuộc loại thứ nhất.

Kết luận

Bài học này đã trình bày chi tiết về hai cấp độ quản lý mã nguồn của C#, bao gồm solution và project.

Bạn nên bỏ thời gian “nghịch” với việc tạo solution, project, đặt không gian tên, cũng như “khám phá” các file trong project của C# để tự tìm hiểu thêm. Nắm bắt rõ các thành phần này sẽ giúp bạn dễ dàng theo dõi các bài học tiếp theo.

- + 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!