Cú pháp C# cơ bản: lệnh, từ khóa, ghi chú, code block, định danh

Bài học này sẽ cung cấp cho bạn những kiến thức cơ bản nhất của C# mà bạn sẽ phải sử dụng thường xuyên trong quá trình học lập trình C#, bao gồm lệnh và khối code (code block), từ khóa, ghi chú, entry point của chương trình C#.

NỘI DUNG CỦA BÀI [Ẩn]

> Để thử nghiệm code trong bài, bạn có thể sử dụng C# interactive cho tiện lợi.

Câu lệnh (statement) và khối code (code block)

Câu lệnh trong C#

Bên trong phương thức Main bạn đã viết ba **câu lệnh** (statement), hai lệnh viết ra màn hình và một lệnh đọc từ bàn phím.

```
Console.WriteLine("Hello world from C#");
Console.WriteLine("Press any key to quit");
Console.ReadKey();
```

Câu lệnh là một hành động chúng ta yêu cầu chương trình thực hiện, ví dụ: khai báo biến, gán giá trị, gọi phương thức, duyệt danh sách, xử lý theo điều kiện giá trị, v.v..

Trình tự thực hiện các lệnh được gọi là **luồng điều khiển** (flow of control) hay **luồng thực thi** (flow of execution).

Một câu lệnh có thể chứa một dòng code duy nhất, gọi là **câu lệnh đơn** (single line statement). C# bắt buộc ***câu lệnh đơn phải kết thúc bằng dấu chấm phẩy***.

Ví dụ, trong thân của phương thức Main() bạn đã viết 3 câu lệnh đơn.

```
Console.WriteLine("Hello world from C#");
Console.WriteLine("Press any key to quit");
Console.ReadKey();
```

Một câu lệnh nếu chỉ chứa dấu chấm phẩy được gọi là một *lệnh rỗng* (empty statement).

Dưới đây là một số loại câu lệnh trong C#:

- Declaration statements: *lệnh khai báo*, dùng để khai báo các biến và hằng;
- Expression statements: *lệnh tính toán*, thường gọi là *biểu thức*, dùng để thực hiện các tính toán trên dữ liệu và phải trả về giá trị có thể gắn cho biến;
- Selection statements: *lệnh lựa chọn*, dùng trong các *cấu trúc rẽ nhánh* như if, else, switch, case;
- Iteration statements: *lệnh lặp*, dùng để thực hiện nhiều lần một lệnh/khối lệnh, bao gồm do, for, foreach, in, while.

Hiện tại chúng ta không giải thích chi tiết các loại lệnh này. Các loại lệnh sẽ lần lượt được xem xét chi tiết khi gặp trong dự án.

Khối code trong C#

Một chuỗi câu lệnh đơn có thể được nhóm lại với nhau tạo thành một **khối lệnh** (code block hoặc statement block).

Một khối lệnh là một danh sách các lệnh được đặt chung trong một cặp dấu ngoặc kép {}.

Các khối lệnh có thể lồng nhau. Như trong file mã nguồn dưới đây, toàn bộ thân của phương thức Main() ở trên là một khối code. Thân của cả lớp Program cũng là một khối code. Thân của namespace là một khối code. Ba khối code này lồng nhau.

Bạn sẽ gặp khối code là thân của namespace, khai báo kiểu (class/struct/enum/interface), khai báo phương thức, cấu trúc điều khiển. Thậm chí có thể có khối code tự do.

Ghi chú (comment) và khoảng trắng

Ghi chú và khoảng trắng là những thông tin không ảnh hưởng đến quá trình dịch. Chúng có nhiệm vụ hỗ trợ người lập trình nhưng sẽ bị compiler bỏ qua khi dịch mã nguồn.

Ghi chú (comment)

Mọi ngôn ngữ lập trình đều cung cấp khả năng ghi lại lời chú thích trực tiếp trong code (mà không ảnh hưởng đến việc dịch code và thực hiện chương trình). Ghi chú dành cho lập trình viên. Compiler sẽ tự động loại bỏ ghi chú khi dịch mã nguồn.

C# cung cấp 3 loại ghi chú khác nhau: ghi chú trên 1 dòng, ghi chú trên nhiều dòng, và ghi chú tài liệu.

Ghi chú trên một dòng sử dụng cặp dấu "//" trước lời ghi chú, những gì đi sau cặp dấu này trên dòng đó được xem là ghi chú.

Ví dụ:

```
Console.WriteLine("Hello world"); // lệnh này viết dòng chữ Hello world ra màn hình
```

Ghi chú trên nhiều dòng sử dụng cặp dấu "/*" ở đầu khối và cặp "*/" ở cuối khối. Ví dụ:

```
/*
Đây là ghi chú trên nhiều dòng.
Lệnh Console.WriteLine sẽ in dòng văn bản ra màn hình.
Lệnh Console.ReadKey sẽ dừng màn hình chờ người dùng ấn một phím bất kỳ trước khi tiếp tục thực thi.
*/
Console.WriteLine("Hello world");
Console.ReadKey();
```

Hai loại ghi chú trên giống như trong C/C++.

> Trong Visual Studio, bạn có thể nhanh chóng đánh dấu chú thích cho nhiều dòng code bằng cách chọn tất cả các dòng cần đánh dấu và ấn tổ hợp phím `Ctrl + K + C`. Để hủy đánh dấu chú thích đã có, chọn các dòng tương ứng và bấm tổ hợp `Ctrl + K + U`.

Ghi chú tài liệu (documentation comment) là loại ghi chú đặc biệt của C#, cho phép tạo ra một dạng "hướng dẫn sử dụng" của đơn vị code được ghi chú (như class, method, interface, v.v.).

Loại ghi chú này cho phép người xây dựng class đưa ra các hướng dẫn cơ bản mà người sử dụng class có thể đọc. Loại ghi chú này cũng cho phép trình biên dịch lọc riêng ra để tạo thành tài liệu hướng dẫn cho code. Ghi chú tài liệu được Visual Studio sinh tự động khi gõ cụm "///" trước đối tượng cần chú thích.

```
1.      /// <summary>
2.      /// class for cars
3.      /// </summary>
4.      class Car
5.      {
6.          // the class body is still empty
7.          // TODO: add more member (field and property)
8.      }
```

Trong ghi chú tài liệu có thể sử dụng các thẻ xml để mô tả nội dung cho các thành phần. Visual Studio sử dụng các thông tin này để tạo ra trợ giúp Intellisense cho đối tượng được chú thích.

Nên hình thành thói quen viết chú thích đầy đủ cho code. Nó giúp ích rất nhiều cho việc bảo trì code hoặc làm việc nhóm.

Khoảng trắng

Khoảng trắng (whitespace) là những ký tự "không nhìn thấy" trong code. Trong C# các ký tự sau được xem là khoảng trắng: ký tự space, tab, ký tự báo dòng mới (new line), ký tự về đầu dòng (carriage return).

Giống như comment, các ký tự trắng trong C# không có giá trị đối với compiler. Nói cách khác, compiler sẽ tự động bỏ qua các ký tự trắng thừa. Tuy nhiên, các ký tự trắng có vai trò rất quan trọng trong định dạng code, giúp code dễ đọc hơn.

Việc sử dụng ký tự trắng để định dạng code được Visual Studio xử lý rất tốt, giúp mã nguồn C# trong Visual studio đặc biệt sáng sủa, dễ đọc. Visual studio đưa ra một loạt các quy tắc định dạng cho từng ngôn ngữ gọi là **code style**. Bạn có thể can thiệp vào các thiết lập này qua cửa sổ Options (Tools => Options), chọn node Text Editor và ngôn ngữ cần điều chỉnh.

Trong quá trình viết code, bạn có thể sử dụng **tổ hợp Ctrl + K + D** để tự động định dạng lại mã nguồn theo các quy ước viết code. Hãy thường xuyên sử dụng tổ hợp này để code được ngăn nắp và quy củ.

Từ khóa (keyword) của C#

File mã nguồn đầu tiên (Program.cs)

Trong ảnh chụp code bạn để ý thấy có một số từ được thể hiện bằng màu xanh dương như *using, namespace, class, static, void, string*. Đây là các **từ khóa** trong C#.

Từ khóa (keyword) là những từ được ngôn ngữ gán cho ý nghĩa riêng xác định, là nòng cốt của cú pháp ngôn ngữ. Bạn không được sử dụng từ khóa cho mục đích gì khác ngoài những gì đã được ngôn ngữ quy định. Ví dụ, bạn không được sử dụng từ khóa làm định danh (tên) của biến, hằng, phương thức.

Dưới đây là danh sách từ khóa trong C#.

danh sách từ khóa của c#

Từ khóa của C# phân chia làm 2 loại: từ khóa dành riêng (reserved keyword) và từ khóa theo ngữ cảnh (contextual keyword). C# có 79 từ khóa dành riêng và 25 từ khóa ngữ cảnh (contextual keyword).

Bạn sẽ dễ dàng nhận thấy từ khóa khi viết code bằng Visual Studio.

> Từ khóa ngữ cảnh là những từ chỉ được xem là từ khóa trong những đoạn code nhất định. Ở những chỗ khác nó không được xem là từ khóa.

Dưới đây là danh sách từ khóa ngữ cảnh của C#:

từ khóa ngữ cảnh trong c#

Lưu ý, bạn không cần ghi nhớ chúng. Đại đa số các từ khóa này chúng ta sẽ gặp lại trong các phần khác nhau của bài giảng.

Định danh (identifier)

Định danh (identifier) là chuỗi ký tự của ngôn ngữ dùng để đặt tên cho các thành phần như kiểu, biến, hằng, phương thức, tham số, v.v..

Quy tắc đặt định danh

Định danh trong C# chỉ được tạo ra từ một nhóm ký tự xác định chứ không được chứa mọi loại ký tự. Nhìn chung, định danh trong C# có thể chứa các chữ cái a-z, A-Z, chữ số 0-9, dấu gạch chân _ và ký tự @.

Tuy vậy, có những giới hạn nhất định.

Ký tự a-z, A-Z và _ có thể có mặt ở mọi vị trí trong định danh.

Chữ số không được phép đứng đầu định danh.

Ký tự @ chỉ được phép đứng đầu định danh (và cũng không được khuyến khích sử dụng).

Khi đặt định danh có sự phân biệt giữa ký tự hoa và thường. Ví dụ write và Write là hai định danh (của phương thức) hoàn toàn khác nhau. Điều này khác biệt với ngôn ngữ như Pascal vốn không phân biệt ký tự hoa và thường.

Quy ước đặt định danh

Ngoài các quy tắc trên, ứng với mỗi loại thành phần có thêm những quy ước đặt định danh. Những quy ước này giúp tạo ra hệ thống tên gọi thống nhất xuyên suốt chương trình, giúp việc bảo trì và đọc code dễ dàng hơn. Khi gặp từng loại thành phần chúng ta sẽ nhắc đến quy ước cụ thể của nó.

Có hai loại quy ước thường gặp: PascalCase và camelCase.

PascalCase là quy ước đặt định danh trong đó:

- Phải bắt đầu bằng chữ cái in hoa;
- Nếu định danh chứa nhiều từ thì các từ đều viết hoa chữ cái đầu tiên.

Bản thân cái tên PascalCase minh họa chính xác các quy ước này: bắt đầu là P (hoa), bao gồm hai từ Pascal và Case thì viết hoa cả P và C. PascalCase thường dùng để đặt tên kiểu (class, struct, enum, interface, delegate), tên phương thức (method), tên các biến/đặc tính thành viên public của struct/class.

Quy ước camelCase hơi khác một chút, trong đó riêng ký tự đầu tiên là chữ cái thường. Chữ cái đầu của các từ tiếp theo viết hoa. Lối viết camelCase thường dùng để đặt tên biến cục bộ, tham số, biến thành viên private của struct/class.

Lớp Program và phương thức Main()

C# là một ngôn ngữ lập trình hướng đối tượng, trong mỗi project bắt buộc phải có ít nhất một lớp. Lớp là đơn vị code quan trọng bậc nhất trong C#, và cả quá trình học ngôn ngữ sau này hầu đều tập trung vào các kỹ thuật xây dựng lớp.

`Program` là lớp duy nhất trong dự án của chúng ta đến giờ, được C# tự động sinh ra khi tạo project. Tên của lớp này không bắt buộc là `Program`. Bạn có thể thay đổi thành bất kỳ tên gọi nào, miễn phù hợp với quy tắc đặt tên.

Trong lớp `Program` có một *phương thức* (method) đặc biệt

```
1.  static void Main(string[] args)
2.  {
3.      Console.WriteLine("Hello world from C#");
4.      Console.WriteLine("Press any key to quit");
5.      Console.ReadKey();
6.  }
```

Phương thức này có tên gọi riêng là "entry point". Main() là phương thức đầu tiên được gọi khi chạy chương trình C#. Đây là điểm khởi đầu trong hoạt động của các chương trình C#.

> Lưu ý rằng, ngôn ngữ C# phân biệt chữ hoa – thường, giống như C/C++. Do đó, Main và main không giống nhau, Writeline khác với WriteLine.

Phương thức entry point của C# bắt buộc phải là `static void Main()` hoặc `static int Main()`. Phần tham số không bắt buộc. Nói chung bạn có thể gặp các biến thể sau của phương thức Main():

```
1.  // int return type, array of strings as the parameter.
2.  static int Main(string[] args)
3.  {
4.      // Must return a value before exiting!
5.      return 0;
6.  }
7.
8.  // No return type, no parameters.
9.  static void Main()
10. {}
11.
12. // int return type, no parameters.
13. static int Main()
14. {
15.     // Must return a value before exiting!
16.     return 0;
17. }
```

> Như bạn đã biết trong bài học giới thiệu về .NET Framework, mỗi project sẽ được dịch thành một assembly, thuộc về một trong hai loại: .exe, hoặc .dll. Phương thức Main bắt buộc phải có trong project dịch ra .exe, nhưng không bắt buộc nếu project dịch ra .dll.

Kết luận

Bài học này đã giới thiệu với bạn những thứ cơ bản nhất, sơ lược nhất, nhưng lại đặc biệt quan trọng trong C#. Nó là nền tảng của mọi thứ "cao cấp" hơn mà bạn sẽ gặp trong bài giảng.

> + 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!