

# Model: class, variable, property, comment, namespace, using

Hướng dẫn tự học lập trình C# toàn tập > Model: class, variable, property, comment, namespace, using

Bài học này sẽ hướng dẫn bạn thực hiện các bước để xây dựng class đầu tiên của dự án: lớp model. Class sẽ xây dựng trong bài đóng vai trò quan trọng và sẽ được sử dụng xuyên suốt các bài thực hành tiếp theo.

Bài học này sẽ dẫn dắt từng bước thực hành để vận dụng các nội dung đã học trong phần lý thuyết. Nội dung chủ yếu của nhóm bài này là vận dụng các kỹ thuật xây dựng `class` sử dụng `biến thành viên` và `property` trong class C#.

## NỘI DUNG CỦA BÀI [ Ấn ]

1. Phân tích các thành phần của class sẽ xây dựng
2. Thực hành 1: xây dựng class Book với biến thành viên
3. Thực hành 2: sử dụng biến public
4. Thực hành 3: sử dụng auto property
5. Thực hành 4: sử dụng full property
6. Thực hành 5: sử dụng documentation comment
7. Thực hành 6: thu gọn khối using, làm gọn code
8. Kết luận

## Phân tích các thành phần của class sẽ xây dựng

Như đã phân tích, một cuốn sách điện tử chứa những thông tin sau:

1. tác giả (hoặc nhóm tác giả),
2. tựa đề,
3. nhà xuất bản,
4. năm xuất bản,
5. lần tái bản,
6. mã ISBN (mã số tiêu chuẩn quốc tế cho sách),
7. mô tả tóm tắt nội dung (dùng cho việc tìm kiếm),
8. từ khóa mô tả nội dung/thể loại,
9. đánh giá của cá nhân (rating, sau này có thể dùng trong sắp xếp),
10. đánh dấu cuốn sách nào hiện đang đọc (để sau dễ dàng tìm đến những cuốn được đánh dấu),
11. đường dẫn đầy đủ tới file sách.

Đây là kết quả của quá trình trừu tượng hóa (phân tích và tách ra các thông tin đặc trưng mà chúng ta quan tâm) từ các cuốn sách về mặt thông tin.

Từ sự trừu tượng hóa đó chúng ta xác định các nội dung cơ bản của class sẽ được xây dựng như bảng dưới đây:

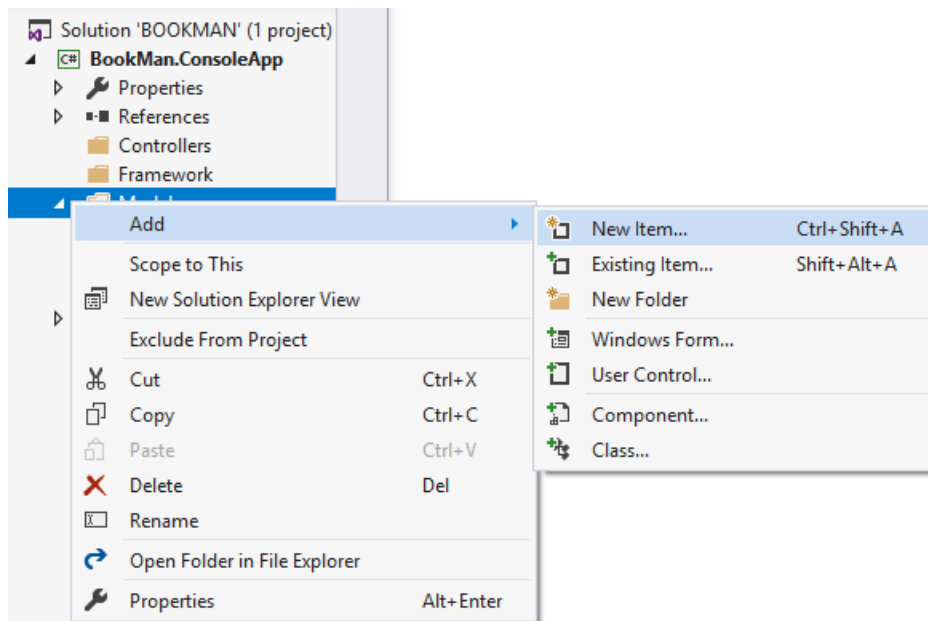
STT	Tên	Mô tả	Kiểu	Ghi chú
1	Id	Số định danh duy nhất	Số nguyên	Bắt buộc
2	Nhóm tác giả (authors)	Danh sách tên tác giả, phân tách bởi dấu phẩy	Văn bản	Bắt buộc
3	Tiêu đề (title)	Tiêu đề sách	Văn bản	Bắt buộc
4	Nhà xuất bản (publisher)	Tên nhà xuất bản	Văn bản	Bắt buộc
5	Năm xuất bản (year)	Năm xuất bản sách	Số nguyên	Bắt buộc, phải lớn hơn 1950, nhỏ hơn năm hiện tại
6	Lần tái bản (edition)	Lần tái bản của sách	Số nguyên	Bắt buộc, phải lớn hơn hoặc bằng 1, mặc định là 1
7	Mã xuất bản (Isbn)	Mã số tiêu chuẩn quốc tế	Văn bản	Không bắt buộc
8	Từ khóa (tags)	Danh sách các từ khóa mô tả nội dung, thể loại	Văn bản	Không bắt buộc
9	Mô tả (description)	Mô tả tóm tắt nội dung	Văn bản	Không bắt buộc
10	Đường dẫn (file)	Đường dẫn (đầy đủ) tới file pdf	Văn bản	Bắt buộc, phải là một đường dẫn đúng
11	Đánh dấu đọc (reading)	Dùng để đánh dấu một cuốn sách đang đọc	Logic	Không bắt buộc, mặc định là false
12	Đánh giá (rating)	Đánh giá chất lượng cuốn sách	Số nguyên	Không bắt buộc, có giá trị từ 1 (dở nhất) đến 5 (tốt nhất), mặc định là 1

Từ đây, chúng ta sẽ xây dựng class đầu tiên với C#. Loại class chỉ chứa thông tin của đối tượng bị quản lý cũng được gọi là *class thực thể* (entity class).

## Thực hành 1: xây dựng class Book với biến thành viên

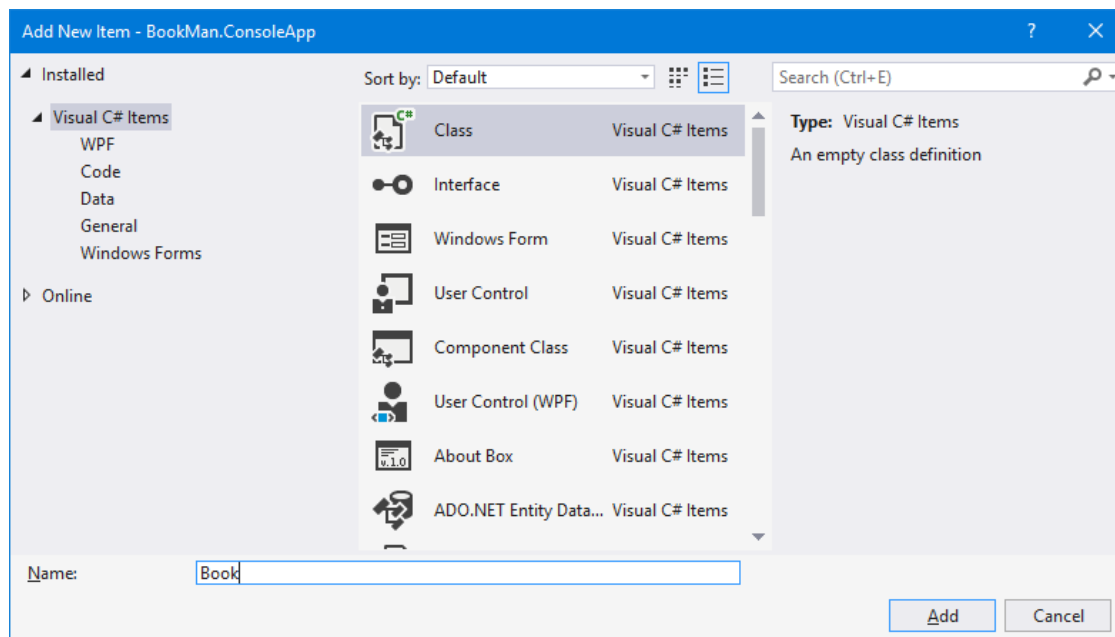
### Bước 1. Tạo file mã nguồn cho class

Click phải vào thư mục Models, chọn Add => New Item ... hoặc bấm tổ hợp Ctrl + Shift + A.



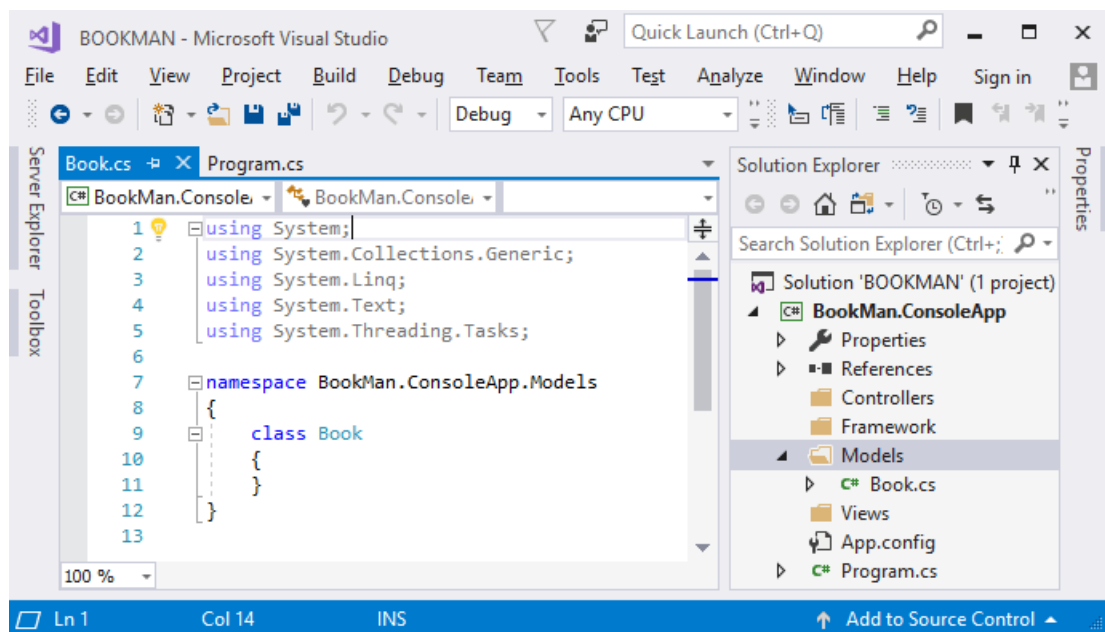
Thêm class mới cho project

Trong hộp thoại "Add New Item" chọn "Class"; trong hộp văn bản "Name" nhập "Book".



Đặt tên file (và tên class)

Bấm nút "Add" hoặc phím "Enter". Một file mã nguồn mới "Book.cs" đã được thêm vào thư mục Models với nội dung như sau:



File mã nguồn của class mới tạo

**Bước 2. Viết code cho lớp Book**

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6.
7. namespace BookMan.ConsoleApp.Models
8. {
9.     class Book
10.    {
11.        int _id;
12.        string _authors;
13.        string _title;
14.        string _publisher;
15.        int _year;
16.        int _edition;
17.        string _isbn;
18.        string _tags;
19.        string _description;
20.        int _rating;
21.        bool _reading;
22.        string _file;
23.    }
24. }
```

Khi xây dựng class Book bạn đã vận dụng quy ước sau:

- đặt tên class kiểu PascalCase;
- đặt toàn bộ code của mỗi class trong một file code riêng cùng tên;
- áp dụng quy ước đặt tên biến thành viên private theo camelCase với dấu \_ ở đầu.

**Thực hành 2: sử dụng biến public**

Trong đoạn code trước, tất cả biến thành viên của " Book " đều được khai báo là " private " và cũng không được gán giá trị đầu. Chúng ta sẽ cải tiến để các trường dữ liệu của " Book " trở thành " public " và gán giá trị đầu phù hợp. Điều chỉnh code của " Book " như sau:

```

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6.
7. namespace BookMan.ConsoleApp.Models
8. {
9.     public class Book
10.    {
11.        public int Id = 1;
12.        public string Authors = "Unknown authors";
13.        public string Title = "A new book";
14.        public string Publisher = "Unknown publisher";
15.        public int Year = 2018;
16.        public int Edition = 1;
17.        public string Ibn;
18.        public string Tags;
19.        public string Description = "A new book";
20.        public int Rating = 1;
21.        public bool Reading = false;
22.        public string File;
23.    }
24. }

```

Với cải tiến này, các object (biến) tạo ra từ kiểu `Book` cho phép truy xuất các trường dữ liệu chứa trong nó và có thể được sử dụng để lưu trữ dữ liệu cho các cuốn sách điện tử đang cần quản lý.

Cũng lưu ý cách đặt tên biến public theo quy ước PascalCase.

Tuy nhiên, cải tiến này có một nhược điểm: các biến thành viên của class này có thể bị truy xuất trực tiếp và chúng ta không kiểm soát được giá trị gán cho chúng.

Ví dụ về yêu cầu giá trị của các trường:

- Id: không nhận giá trị nhỏ hơn 1;
- Authors, Title, Publisher: không được nhận chuỗi rỗng;
- Edition: không được nhận các giá trị nhỏ hơn 1;
- Rating: chỉ nhận các giá trị từ 1 đến 5, tương đương với các mức đánh giá từ xấu đến tốt;
- File: chỉ chấp nhận đường dẫn chính xác tới file sách pdf.

## Thực hành 3: sử dụng auto property

Thay đổi code của class `Book` như sau:

```

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace BookMan.ConsoleApp.Models
7. {
8.     public class Book
9.    {
10.        public int Id { get; set; } = 1;
11.        public string Authors { get; set; } = "Unknown author";
12.        public string Title { get; set; } = "A new book";
13.        public string Publisher { get; set; } = "Unknown publisher";
14.        public int Year { get; set; } = 2018;
15.        public int Edition { get; set; } = 1;

```

```
16.     public string Isbn { get; set; }
17.     public string Tags { get; set; }
18.     public string Description { get; set; } = "A new book";
19.     public int Rating { get; set; } = 1;
20.     public bool Reading { get; set; }
21.     public string File { get; set; }
22. }
23. }
```

Lưu ý cách đặt tên property theo quy ước PascalCase.

Bạn có thể sử dụng Code snippet để nhanh chóng tạo ra property theo cách sau:

1. Nhập cụm "prop" => trong danh sách lựa chọn của Visual Studio IntelliSense xuất hiện mục "prop" => chọn mục này và bấm phím Tab hai lần.
2. Trong snippet vừa tạo chỉnh sửa các thông tin cần thiết (phần code được bôi vàng). Di chuyển giữa các vùng bôi vàng bằng phím Tab. Kết thúc chỉnh sửa snippet bằng phím Enter.

So sánh với code cũ bạn có thể thấy sau tên của mỗi trường dữ liệu xuất hiện một khối code { get; set; }. Phần thực hành này hoàn toàn sử dụng [auto property](#).

## Thực hành 4: sử dụng full property

Như đã phân tích, trong class "Book" có một số giới hạn cần đặt ra với dữ liệu:

- Id không nhận giá trị nhỏ hơn 1;
- Authors, Title, Publisher không được nhận chuỗi rỗng;
- Year không nhận các giá trị nhỏ hơn 1950 (sách cũ quá!);
- Edition không nhận giá trị nhỏ hơn 1;
- Rating chỉ nhận giá trị trong khoảng [1, 5];
- File chỉ được nhận giá trị là đường dẫn chính xác tới file.

Ngoài ra, chúng ta cũng muốn tạo ra một trường trong đó chứa tên ngắn gọn của File sách (để tiện sử dụng trong một số trường hợp).

Chúng ta thay đổi code của class "Book" như sau:

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace BookMan.ConsoleApp.Models
7. {
8.     public class Book
9.     {
10.         private int _id = 1;
11.         public int Id
12.         {
13.             get { return _id; }
14.             set { if (value >= 1) _id = value; }
15.         }
16.         private string _authors = "Unknown author";
17.         public string Authors
18.         {
19.             get { return _authors; }
20.             set { if (!string.IsNullOrEmpty(value)) _authors = value; }
21.         }
22.         private string _title = "A new book";
23.         public string Title
24.         {
25.             get { return _title; }
26.             set { if (!string.IsNullOrEmpty(value)) _title = value; }
27.         }
28.         private string _publisher = "Unknown publisher";
29.         public string Publisher
30.         {
31.             get { return _publisher; }
32.             set { if (!string.IsNullOrEmpty(value)) _publisher = value; }
33.         }
34.         private int _year = 2018;
35.         public int Year
36.         {
37.             get { return _year; }
38.             set { if (value >= 1950) _year = value; }
39.         }
40.         private int _edition = 1;
41.         public int Edition
42.         {
43.             get { return _edition; }
44.             set { if (value >= 1) _edition = value; }
45.         }
46.         public string Isbn { get; set; } = "";
47.         public string Tags { get; set; } = "";
48.         public string Description { get; set; } = "A new book";
49.         private int _rating = 1;
50.         public int Rating
```

```

51.         {
52.             get { return _rating; }
53.             set { if (value >= 1 && value <= 5) _rating = value; }
54.         }
55.     public bool Reading { get; set; }
56.     private string _file;
57.     public string File
58.     {
59.         get { return _file; }
60.         set { if (System.IO.File.Exists(value)) _file = value; }
61.     }
62.     public string FileName
63.     {
64.         get { return System.IO.Path.GetFileName(_file); }
65.     }
66. }
67. }

```

Như vậy, phần thực hành này đã vận dụng **full property** để kiểm soát dữ liệu nhập vào cho các trường backed field.

## Thực hành 5: sử dụng documentation comment

Các bạn đã biết, C# sử dụng một loại chú thích đặc biệt gọi là **chú thích tài liệu** (documentation comment). Loại chú thích này rất có ích cho người lập trình vì nó giúp mô tả các đơn vị code như **class**, **method**, **interface**, v.v. xuyên suốt trong project.

Phần thực hành này sẽ bổ sung các chú thích cần thiết cho lớp Book.

```

1.     using System;
2.     using System.Collections.Generic;
3.     using System.Linq;
4.     using System.Text;
5.     using System.Threading.Tasks;
6.     namespace BookMan.ConsoleApp.Models
7.     {
8.         /// <summary>
9.         /// Lớp mô tả sách điện tử
10.        /// </summary>
11.        public class Book
12.        {
13.            private int _id = 1;
14.            /// <summary>
15.            /// số định danh duy nhất cho mỗi object
16.            /// </summary>
17.            public int Id
18.            {
19.                get { return _id; }
20.                set { if (value >= 1) _id = value; } // id chỉ nhận giá trị >= 1
21.            }
22.            private string _authors = "Unknown author";
23.            /// <summary>
24.            /// tên tác giả hoặc nhóm tác giả, không nhận xâu rỗng
25.            /// </summary>
26.            public string Authors
27.            {
28.                get { return _authors; }
29.                set { if (!string.IsNullOrEmpty(value)) _authors = value; } // không nhận xâu
30.            }
31.            private string _title = "A new book";
32.            /// <summary>
33.            /// tiêu đề sách, không nhận xâu rỗng
34.            /// </summary>
35.            public string Title
36.            {
37.                get { return _title; }
38.                set { if (!string.IsNullOrEmpty(value)) _title = value; } // không nhận xâu
39.            }
40.            private string _publisher = "Unknown publisher";
41.            /// <summary>
42.            /// nhà xuất bản, không nhận xâu rỗng
43.            /// </summary>

```



```

44.     public string Publisher
45.     {
46.         get { return _publisher; }
47.         set { if (!string.IsNullOrEmpty(value)) _publisher = value; } // không nhận
48.     }
49.     private int _year = 2018;
50.     /// <summary>
51.     /// năm xuất bản, không nhỏ hơn 1950
52.     /// </summary>
53.     public int Year
54.     {
55.         get { return _year; }
56.         set { if (value >= 1950) _year = value; } // năm không nhỏ hơn 1950
57.     }
58.     private int _edition = 1;
59.     /// <summary>
60.     /// lần tái bản, không nhỏ hơn 1
61.     /// </summary>
62.     public int Edition
63.     {
64.         get { return _edition; }
65.         set { if (value >= 1) _edition = value; } // không nhận giá trị < 1
66.     }
67.     /// <summary>
68.     /// mã số quốc tế
69.     /// </summary>
70.     public string Isbn { get; set; } = "";
71.     /// <summary>
72.     /// từ khóa mô tả nội dung / thể loại
73.     /// </summary>
74.     public string Tags { get; set; } = "";
75.     /// <summary>
76.     /// mô tả tóm tắt nội dung
77.     /// </summary>
78.     public string Description { get; set; } = "A new book";
79.     private int _rating = 1;
80.     /// <summary>
81.     /// đánh giá cá nhân, giá trị từ 1 đến 5
82.     /// </summary>
83.     public int Rating
84.     {
85.         get { return _rating; }
86.         set { if (value >= 1 && value <= 5) _rating = value; } // giá trị từ 1 đến 5
87.     }
88.     /// <summary>
89.     /// đánh dấu là đang đọc
90.     /// </summary>
91.     public bool Reading { get; set; }
92.     private string _file;
93.     /// <summary>
94.     /// file sách (gồm đường dẫn)
95.     /// </summary>
96.     public string File
97.     {
98.         get { return _file; }
99.         set { if (System.IO.File.Exists(value)) _file = value; } // nhận đường dẫn đ
100.    }
101.    /// <summary>
102.    /// file sách (không có đường dẫn)
103.    /// </summary>
104.    public string FileName
105.    {
106.        get { return System.IO.Path.GetFileName(_file); } // trả lại tên file ngắn g
107.    }
108.    }
109. }

```

Khi class (và kiểu dữ liệu nói chung), property hoặc phương thức có chú thích tài liệu, nếu đặt con trỏ chuột lên trên sẽ xuất hiện thông tin này, giúp người lập trình (bản thân hoặc người khác) dễ dàng hiểu được tác dụng của các đơn vị code này.

Mỗi loại đối tượng trong C# đều được Visual Studio hiển thị bằng một biểu tượng riêng. Như class, method, property, variable đều được biểu thị bằng một biểu tượng riêng, giúp người lập trình dễ dàng nhận biết đây là loại đối tượng nào.

Biểu tượng kết hợp với chú thích tài liệu hỗ trợ rất tốt cho việc tự tìm hiểu code.

Dưới đây là ví dụ đối với class “Book” và thuộc tính “Authors”. Trong hình là cách Visual Studio hiển thị ghi chú tài liệu khi đặt con trỏ chuột vào tên class Book và thuộc tính Authors. Cũng lưu ý biểu tượng của class và property.

*Biểu tượng class trong Visual Studio*

*Biểu tượng property trong Visual Studio*

Nên tập thành thói quen ghi chú code. Đối với các khai báo kiểu, property, method thì nên dùng documentation comment. Trong thân phương thức thì dùng comment thông thường.

## Thực hành 6: thu gọn khối using, làm gọn code

Sau khi hoàn thành một class có thể xóa bỏ những mục `using` không sử dụng tới để file code gọn gàng hơn. Visual Studio 2017 hỗ trợ tính năng thông báo này bằng cách hiện màu nhạt cho những mục using thừa (không sử dụng class nào trong `namespace` tương ứng). Xóa bỏ những mục này là an toàn và giúp code nhìn gọn gàng hơn.

Visual Studio cung cấp một Quick Action để "thu dọn" khối using: đặt con trỏ vào vùng using (chỗ hiện màu nhạt); kích hoạt Quick Action (tổ hợp Ctrl + .); chọn Remove Unnecessary Usings.

*Sử dụng quick action để dọn dẹp khối using*

Để format lại toàn bộ code theo quy ước viết code chung của Visual Studio, bạn bấm tổ hợp Ctrl + K + D.

Đến đây xin chúc mừng bạn đã xây dựng được một class đầu tiên “theo kiểu C#”.

## Kết luận

---

Bài học này giúp bạn tổng hợp và vận dụng các kiến thức đã học về cách xây dựng class cơ bản trong C#. Nội dung chủ yếu của nhóm bài này là kỹ thuật xây dựng class sử dụng **biến thành viên** và **property** trong class C#.

Khi vận dụng các kỹ thuật này một cách phù hợp, bạn đã tạo ra được một class thực thể Book một cách “chuyên nghiệp” “theo kiểu C#”.

- + 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!