**Bài Thực Hành Tuần 2 (LNNet)**

**1/ Xây dựng lớp Book**

*Yêu cầu*:

Xây dựng lớp Book với các thuộc tính Author, Title, PublishedDate, và Price. Trong đó mỗi thuộc tính đều có phương thức gán và lấy giá trị thông qua get và set.

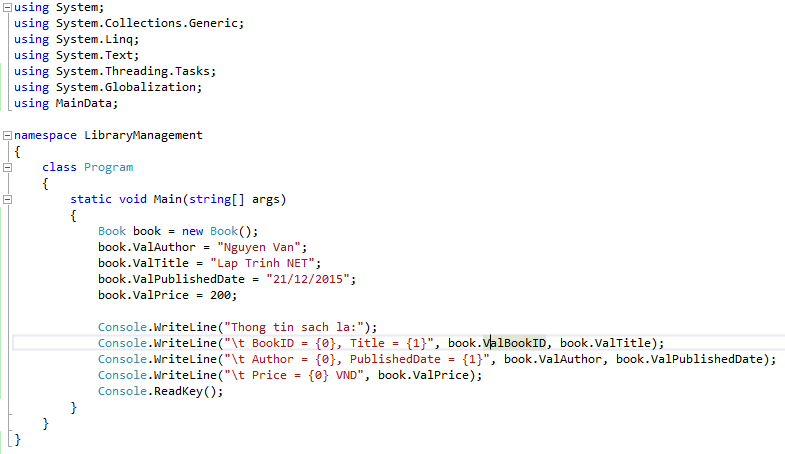
*Hướng dẫn*:

* Tạo một namespace riêng tên là MainData để chứa lớp Book
* Khai báo các thuộc tính Author, Title, PublishedDate, và Price và phương thức get set tương ứng
* Viết phương thức thiết lập có tham số mặc định
* Dùng kiểu DateTime để hiển thị thông tin PublishedDate. Dùng lệnh sau để chuyển một chuỗi thành DateTime

Datetime PublishedDate = DateTime.ParseExact(“21/12/2016”,"dd/MM/yyyy",CultureInfo.InvariantCulture);

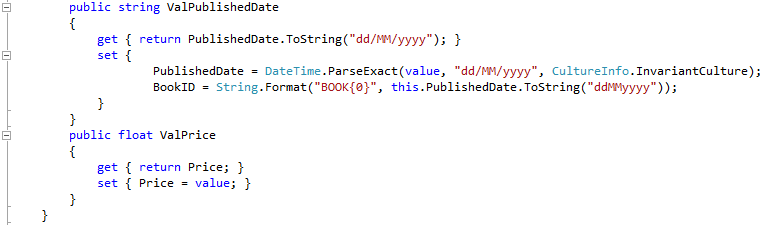
(Lưu ý để dùng CultureInfo thì phải có using System.Globalization;)

* Viết hàm main tạo một book và xuất thông tin book đó ra









**2/ Hiển thị MessageBox trong màn hình Console**

*Yêu cầu*:

Hiển thị thông tin sách thông qua một hộp thoại trên màn hình Console.

*Hướng dẫn*:

Dùng thuộc tính DllImport để import User32.dll (dll này đã có sẵn) để hiển thị hộp thoại trên màn hình console.

* Sử dụng namespace: using System.Runtime.InteropServices;
* Khai báo thuộc tính DllImport để thông báo sẽ sử dụng dll nào và khai báo lệnh sẽ sử dụng

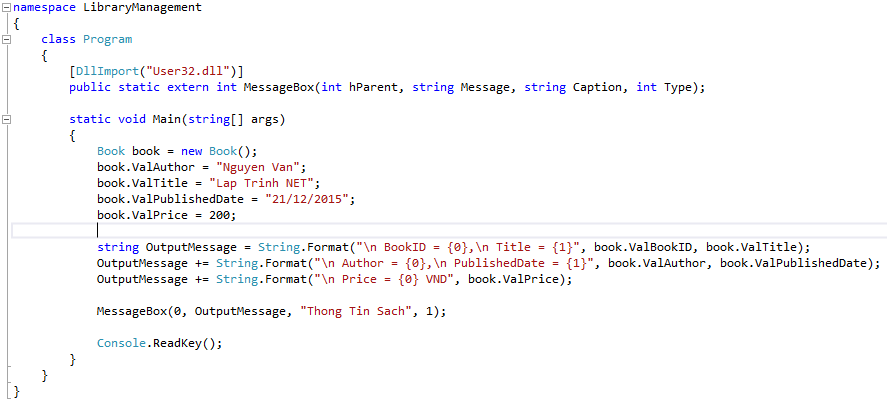
[DllImport("User32.dll")]

public static extern int MessageBox(int hParent, string Message, string Caption, int Type);

* Trong hàm main gọi MessageBox để hiển thị hộp thoại

MessageBox(0, OutputMessage, "Thong Tin Sach", 1);





**3/ Xây dựng thuộc tính CodingInfo ghi chú các thông tin coding**

*Yêu cầu*:

Xây dựng thuộc tính CodingInfo để cho ghi chú các thông tin DeveloperName, CreatedDate, ModifiedDate, Comment cho biết ai và khi nào Lớp Book được tạo và chỉnh sửa.

*Hướng dẫn*:

Tạo một namespace mới tên AttributeData và lớp CodingInfo kế thừa từ System.Attribute. Lớp CodingInfo sẽ đóng vai trò như một Attribute cung cấp thông tin về DeveloperName, CreatedDate, ModifidedDate và Comment

* Tạo namespace AttributeData
* Tạo lớp CodingInfo kế thừa từ System.Attribute
* Thêm các thuộc tính DeveloperName, CreatedDate, ModifidedDate và Comment
* Thêm phương thức khởi tạo có tham số
* Thêm dòng sau vào đầu lớp CodingInfo

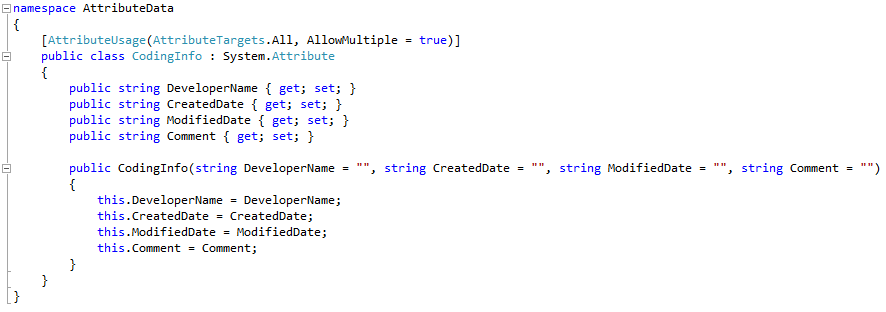
[AttributeUsage(AttributeTargets.All, AllowMultiple = true)]

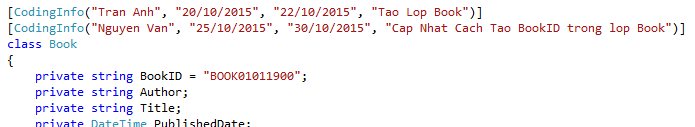
* Thêm các dòng ghi chú vào đầu lớp Book cho biết thông tin ai tạo, ai chỉnh sửa

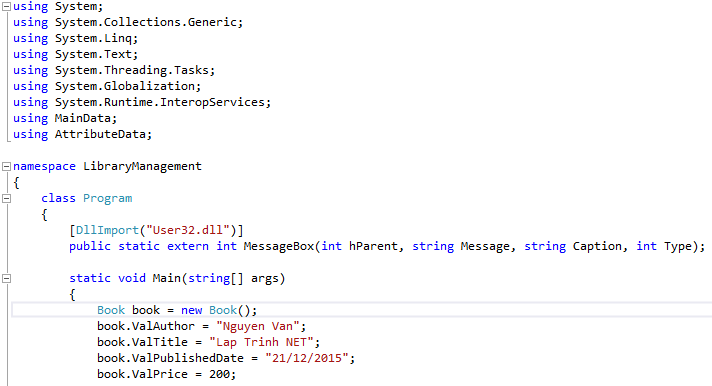
[CodingInfo("Tran Anh", "20/10/2015", "22/10/2015", "Tao Lop Book")]

[CodingInfo("Nguyen Van", "25/10/2015", "30/10/2015", "Cap Nhat Cach Tao BookID trong lop Book")]

* Thêm dòng using AttributeData;
* Viết hàm main để đọc các thông tin chỉnh sửa lớp Book









**4/ Tạo interface và sử dụng interface**

*Yêu cầu*:

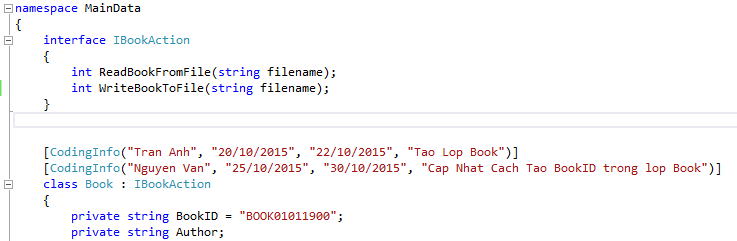
Tạo một interface với hai chức năng là đọc và ghi dữ liệu book qua file text.

*Hướng dẫn*:

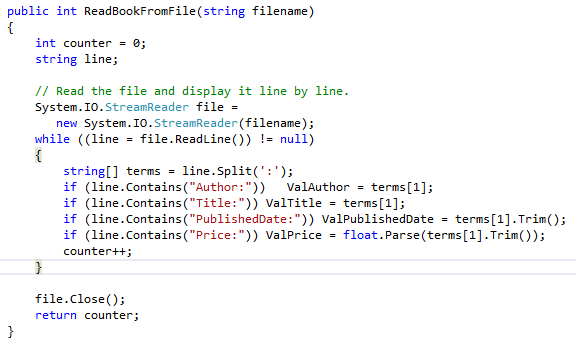
* Trong namespace MainData tạo một interface tên IBookAction với hai phương thức đọc và ghi file
* Cho lớp book kế thừa và sử dụng interface đó bằng cách khai báo

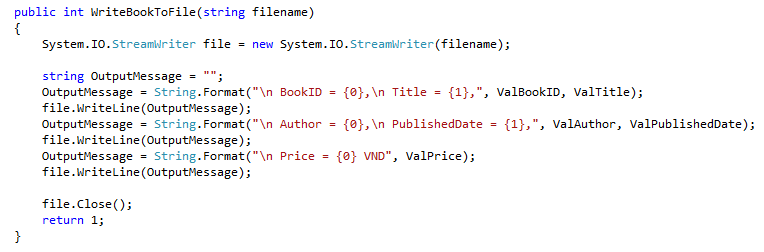
class Book : IBookAction

* Sau đó định nghĩa cụ thể hai phương thức đọc và ghi file này. Dùng StreamReader và StreamWriter trong namespace System.IO để đọc và ghi file
* Trong hàm main thực hiện gọi hai phương thức đọc và ghi file này

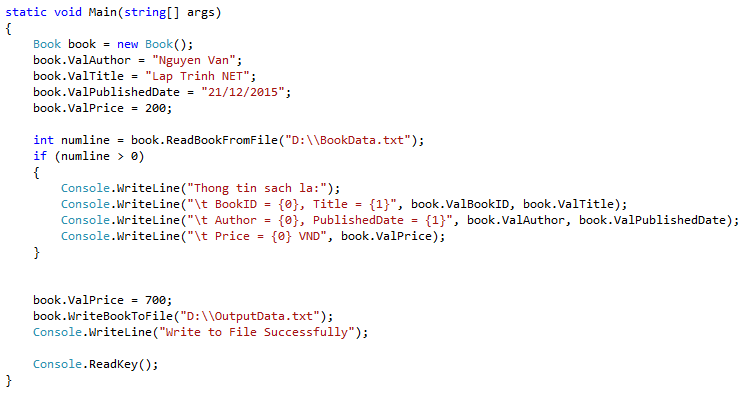


Trong lớp Book:





Hàm Main



**5/ Dùng delegate để thực thi các câu lệnh**

*Yêu cầu*:

Viết một delegate BookReadWrite để thực thi các lệnh đọc và ghi file từ lớp Book

*Hướng dẫn*:

* Khai báo một delegate BookReadWrite bên ngoài các namespaces

delegate int BookReadWrite(string filename);

* Hàm main khai báo hai đối tượng action1 và action2 thuộc kiểu delegate này và tham chiếu lần lượt đến các hàm đọc và ghi file trong lớp Book
* Gọi Invoke để yêu cầu action1 và action2 thực thi các thao tác đọc và ghi mong muốn

