

View cập nhật dữ liệu (2): nạp chồng, phương thức mở rộng

[Hướng dẫn tự học lập trình C# toàn tập](#) > [View cập nhật dữ liệu \(2\): nạp chồng, phương thức mở rộng](#)

Trong bài học này chúng ta sẽ học thêm kỹ thuật [nạp chồng phương thức](#) (method overloading) và [phương thức mở rộng](#) (extension method) của C#. Chúng ta sẽ vận dụng các kỹ thuật này để tiếp tục cải tiến lớp view cập nhật thông tin.

NỘI DUNG CỦA BÀI [Ẩn]

- Thực hành 1: cải tiến tiếp lớp ViewHelp và BookUpdateView
 - Bước 1. Bổ sung phương thức vào lớp ViewHelp
 - Bước 2. Cải tiến lớp BookUpdateView
 - Bước 3. Dịch và chạy thử với lệnh update
- Thực hành: tiếp tục cải tiến lớp BookUpdateView
 - Bước 1. Xây dựng lớp Extension
 - Bước 2. Bổ sung thêm phương thức vào lớp ViewHelp
 - Bước 3. Cập nhật phương thức Render của lớp BookUpdateView
 - Bước 4. Dịch và chạy thử chương trình với lệnh update
- Kết luận

Thực hành 1: cải tiến tiếp lớp ViewHelp và BookUpdateView

Bước 1. Bổ sung phương thức vào lớp ViewHelp

Bổ sung phương thức sau đây vào cuối class ViewHelp

```
1.  /// <summary>
2.  /// cập nhật giá trị kiểu string. Nếu ấn enter mà không nhập dữ liệu sẽ trả lại giá trị
3.  /// </summary>
4.  /// <param name="label">dòng thông báo</param>
5.  /// <param name="oldValue">giá trị gốc</param>
6.  /// <param name="labelColor">màu chữ thông báo</param>
7.  /// <param name="valueColor">màu chữ dữ liệu</param>
8.  /// <returns></returns>
9.  public static string InputString(string label, string oldValue, ConsoleColor labelColor
10. {
11.     Write($"{label}: ", labelColor);
12.     WriteLine(oldValue, ConsoleColor.Yellow);
13.     Write("New value >> ", ConsoleColor.Green);
14.     Console.ForegroundColor = valueColor;
15.     string newValue = Console.ReadLine();
16.     return string.IsNullOrEmpty(newValue.Trim()) ? oldValue : newValue;
17. }
```

Đây là một overload khác của phương thức InputString mà bạn đã tạo từ các bài học trước. Overload này có nhiệm vụ cập nhật giá trị cho một biến kiểu string.

Phương thức này cập nhật một chuỗi ký tự từ console theo logic sau:

1. viết dòng thông báo để người dùng nhập dữ liệu (tham số label);
2. nếu người dùng nhập một chuỗi ký tự thì chuỗi mới sẽ được trả lại;
3. nếu ấn enter luôn (tức là nhập chuỗi trống) thì trả lại giá trị cũ (tham số oldValue).

Bước 2. Cải tiến lớp BookUpdateView

Điều chỉnh phương thức Render để sử dụng các phương thức InputString vừa tạo.

```
1.  using Framework;
2.  using System;
3.
4.  namespace BookMan.ConsoleApp.Views
5.  {
6.      using Models;
7.
8.      internal class BookUpdateView
9.      {
10.         protected Book Model;
11.
12.         public BookUpdateView(Book model)
13.         {
14.             Model = model;
15.         }
16.
17.         public void Render()
18.         {
19.             ViewHelp.WriteLine("UPDATE BOOK INFORMATION", ConsoleColor.Green);
20.
21.             var authors = ViewHelp.InputString("Authors", Model.Authors);
22.             var title = ViewHelp.InputString("Title", Model.Title);
23.             var publisher = ViewHelp.InputString("Publisher", Model.Publisher);
24.             var isbn = ViewHelp.InputString("Isbn", Model.Isbn);
25.             var tags = ViewHelp.InputString("Tags", Model.Tags);
26.             var description = ViewHelp.InputString("Description", Model.Description);
27.             var file = ViewHelp.InputString("File", Model.File);
28.
29.             // TẠM DỪNG ....
30.         }
31.     }
32. }
```

Ở bước này chúng ta sử dụng phương thức `InputString` của lớp `ViewHelp` để cập nhật dữ liệu cho các trường thuộc kiểu `string` của lớp `Book`.

Bước 3. Dịch và chạy thử với lệnh update

Kết quả hoạt động của chương trình

Trong phần thực hành trên chúng ta viết thêm một phương thức `InputString` nữa trong class `ViewHelp`. Lưu ý rằng, trong class đã có sẵn một phương thức có tên `InputString`. Đây là *nạp chồng phương thức* (method overloading).

Khi sử dụng phương thức có nạp chồng, Intellisense của Visual Studio sẽ hiển thị số lượng các phiên bản nạp chồng và cho phép lựa chọn phiên bản muốn sử dụng.

Cách Visual Studio hiển thị các phương thức nạp chồng

Hai phương thức `InputString` trong lớp `ViewHelp` có nguyên mẫu (prototype) lần lượt như sau:

```
1. string InputString(string label, ConsoleColor labelColor = ConsoleColor.Magenta, Console
2.
3. string InputString(string label, string oldValue, ConsoleColor labelColor = ConsoleColor
```

Như vậy, danh sách tham số của hai phương thức này là khác nhau (trước hết là về số lượng).

Thực hành: tiếp tục cải tiến lớp `BookUpdateView`

Bước 1. Xây dựng lớp `Extension`

Thêm file mã nguồn mới `Extension.cs` trong thư mục `Framework`. Trong file này định nghĩa một class `Extension` như sau:

```
1. namespace Framework
2. {
3.     /// <summary>
4.     /// Một số phương thức mở rộng để biến đổi kiểu dữ liệu
5.     /// </summary>
6.     public static class Extension
```

```

7.  {
8.      /// <summary>
9.      /// Biến đổi từ chuỗi sang số nguyên
10.     /// </summary>
11.     /// <param name="value"></param>
12.     /// <returns></returns>
13.     public static int ToInt(this string value)
14.     {
15.         return int.Parse(value);
16.     }
17.
18.     /// <summary>
19.     /// Biến đổi từ chuỗi sang số nguyên
20.     /// </summary>
21.     /// <param name="value"></param>
22.     /// <param name="i"></param>
23.     /// <returns></returns>
24.     public static bool ToInt(this string value, out int i)
25.     {
26.         return int.TryParse(value, out i);
27.     }
28.
29.     /// <summary>
30.     /// Biến đổi chuỗi Y,y,True, true, sang true
31.     /// Các chuỗi khác thành false
32.     /// </summary>
33.     /// <param name="value"></param>
34.     /// <returns></returns>
35.     public static bool ToBool(this string value)
36.     {
37.         var v = value.ToLower();
38.         if (v == "y" || v == "true") return true;
39.         return false;
40.     }
41.
42.     /// <summary>
43.     /// Biến đổi true/false thành Yes/No hoặc có/không
44.     /// </summary>
45.     /// <param name="value"></param>
46.     /// <param name="format">y/n hoặc c/k</param>
47.     /// <returns></returns>
48.     public static string ToString(this bool value, string format)
49.     {
50.         if (format == "y/n")
51.             return value ? "Yes" : "No";
52.         if (format == "c/k")
53.             return value ? "Có" : "Không";
54.         return value ? "True" : "False";
55.     }
56. }
57. }

```

Bước 2. Bổ sung thêm phương thức vào lớp ViewHelp

Bổ sung thêm các overload sau vào lớp ViewHelp:

```

1.  public static int InputInt(string label, int oldValue, ConsoleColor labelColor = Console
2.  {
3.      Write($"{label}: ", labelColor);
4.      WriteLine($"{oldValue}", ConsoleColor.Yellow);
5.      Write("New value >> ", ConsoleColor.Green);
6.      Console.ForegroundColor = valueColor;
7.      string str = Console.ReadLine();
8.      if (string.IsNullOrEmpty(str)) return oldValue;
9.      if (str.ToInt(out int i)) return i; //sử dụng phương thức mở rộng ToInt
10.     return oldValue;
11. }
12.
13. public static bool InputBool(string label, bool oldValue, ConsoleColor labelColor = Cons
14. {
15.     Write($"{label}: ", labelColor);
16.     //sử dụng phương thức mở rộng ToString
17.     WriteLine(oldValue.ToString("y/n"), ConsoleColor.Yellow);
18.     Write("New value >> ", ConsoleColor.Green);
19.     Console.ForegroundColor = valueColor;
20.     string str = Console.ReadLine();
21.     if (string.IsNullOrEmpty(str)) return oldValue;
22.     return str.ToBool(); //sử dụng phương thức mở rộng ToBool

```

```
23. | }  
◀ ▶
```

Bước 3. Cập nhật phương thức Render của lớp BookUpdateView

```
1.  using Framework;  
2.  using System;  
3.  
4.  namespace BookMan.ConsoleApp.Views  
5.  {  
6.      using Models;  
7.  
8.      internal class BookUpdateView  
9.      {  
10.         protected Book Model;  
11.  
12.         public BookUpdateView(Book model)  
13.         {  
14.             Model = model;  
15.         }  
16.  
17.         public void Render()  
18.         {  
19.             ViewHelp.WriteLine("UPDATE BOOK INFORMATION", ConsoleColor.Green); //sử dụng  
20.  
21.             var authors = ViewHelp.InputString("Authors", Model.Authors);  
22.             var title = ViewHelp.InputString("Title", Model.Title);  
23.             var publisher = ViewHelp.InputString("Publisher", Model.Publisher);  
24.             var isbn = ViewHelp.InputString("Isbn", Model.Isbn);  
25.             var tags = ViewHelp.InputString("Tags", Model.Tags);  
26.             var description = ViewHelp.InputString("Description", Model.Description);  
27.             var file = ViewHelp.InputString("File", Model.File);  
28.             var year = ViewHelp.InputInt("Year", Model.Year);  
29.             var edition = ViewHelp.InputInt("Edition", Model.Edition);  
30.             var rating = ViewHelp.InputInt("Rate", Model.Rating);  
31.             var reading = ViewHelp.InputBool("Reading", Model.Reading);  
32.         }  
33.     }  
34. }
```

Bước 4. Dịch và chạy thử chương trình với lệnh update

Kết quả thực hiện chương trình

Trong bước 1 của thực hành trên chúng ta gặp bốn phương thức trong static class Extension:

```
1. public static int ToInt(this string value)
2. {
3.     return int.Parse(value);
4. }
5. public static bool ToInt(this string value, out int i)
6. {
7.     return int.TryParse(value, out i);
8. }
9. public static bool ToBool(this string value)
10. {
11.     var v = value.ToLower();
12.     if (v == "y" || v == "true") return true;
13.     return false;
14. }
15. public static string ToString(this bool value, string format)
16. {
17.     if (format == "y/n")
18.         return value ? "Yes" : "No";
19.     if (format == "c/k")
20.         return value ? "Có" : "Không";
21.     return value ? "True" : "False";
22. }
```

Đây là các phương thức thuộc loại **phương thức mở rộng** (extension method).

Kết luận

Trong bài này chúng ta đã xây dựng được một class nữa để cập nhật thông tin sách. Qua đó, chúng ta làm quen với việc sử dụng phương thức mở rộng và nạp chồng phương thức.

Chúng ta cũng sử dụng các công cụ này để cải tiến các lớp view đã xây dựng ở các bài trước giúp loại bỏ tình trạng lặp code giữa các class.

- + 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!