

View nhập dữ liệu: biến cục bộ, switch-case, biến đổi kiểu, tham số

Hướng dẫn tự học lập trình C# toàn tập > View nhập dữ liệu: biến cục bộ, switch-case, biến đổi kiểu, t...

Trong bài này chúng ta tiếp tục xây dựng một lớp view nữa để nhập thông tin từ người dùng. Qua bài này chúng ta sẽ tiếp xúc với một số vấn đề: [biến cục bộ](#), nhập dữ liệu từ [console](#), [cấu trúc điều khiển](#), [biến đổi kiểu](#), [tham số của phương thức](#) (tham số tùy chọn, tham số out).

NỘI DUNG CỦA BÀI [Ẩn]

- Thực hành 1: xây dựng class giúp nhập thông tin của một cuốn sách mới
 - Bước 1. Thêm lớp BookCreateView
 - Bước 3. Bổ sung phương thức Create cho BookController
- Thực thi lệnh theo yêu cầu của người dùng
- Thực hành 2: nhận lệnh từ người dùng
 - Bước 1. Chính sửa code của phương thức Main
 - Bước 2. Dịch và chạy thử chương trình với lệnh single
 - Bước 3. Bổ sung khả năng thực hiện lệnh create
 - Bước 4. Dịch và chạy thử chương trình với lệnh create
- Thực hành 3: cải tiến BookCreateView
 - Bước 1. Bổ sung phương thức in ra console với màu sắc
 - Bước 2. Sửa code của phương thức Render
- Thực hành 4: tiếp tục cải tiến BookCreateView
 - Bước 1. Xây dựng phương thức nhập dữ liệu từ Console
 - Bước 2. Cải tiến code của phương thức Render
 - Bước 3. Dịch và chạy thử chương trình
- Kết luận

Thực hành 1: xây dựng class giúp nhập thông tin của một cuốn sách mới

Trong các bài trước chúng ta đã xây dựng lớp view đầu tiên để hiển thị dữ liệu ra giao diện console. Chúng ta sẽ tiếp tục với lớp view thứ hai giúp nhập thông tin từ console.

Bước 1. Thêm lớp BookCreateView

Trong thư mục Views thêm file mã nguồn mới (BookCreateView.cs) cho lớp

BookCreateView

Nhập code cho lớp BookCreateView như sau:

```
1. using System;
2.
3. namespace BookMan.ConsoleApp.Views
4. {
5.     /// <summary>
6.     /// class để thêm một cuốn sách mới
7.     /// </summary>
8.     internal class BookCreateView
9.     {
10.         public BookCreateView()
11.         {
12.
```

```

13.     }
14.
15.     /// <summary>
16.     /// yêu cầu người dùng nhập từng thông tin và lưu lại thông tin đó
17.     /// </summary>
18.     public void Render()
19.     {
20.         Console.ForegroundColor = ConsoleColor.Green;
21.         Console.WriteLine("CREATE A NEW BOOK");
22.         Console.ResetColor();
23.
24.         Console.ForegroundColor = ConsoleColor.Magenta;
25.         Console.Write("Title: ");
26.         Console.ResetColor();
27.         string title = Console.ReadLine(); //đọc 1 dòng và lưu vào biến title
28.
29.         Console.ForegroundColor = ConsoleColor.Magenta;
30.         Console.Write("Authors: ");
31.         Console.ResetColor();
32.         string authors = Console.ReadLine(); //đọc 1 dòng và lưu vào biến authors
33.
34.         Console.ForegroundColor = ConsoleColor.Magenta;
35.         Console.Write("Publisher: ");
36.         Console.ResetColor();
37.         string publisher = Console.ReadLine(); //đọc 1 dòng và lưu vào biến publisher
38.
39.         Console.ForegroundColor = ConsoleColor.Magenta;
40.         Console.Write("Year: ");
41.         Console.ResetColor();
42.         string yearString = Console.ReadLine(); //đọc 1 dòng và lưu vào biến yearStr
43.         int year = int.Parse(yearString); //chuyển đổi chuỗi sang số nguyên
44.
45.         Console.ForegroundColor = ConsoleColor.Magenta;
46.         Console.Write("Edition: ");
47.         Console.ResetColor();
48.         string editionString = Console.ReadLine(); //đọc 1 dòng và lưu vào biến edit
49.         int edition = int.Parse(editionString); //chuyển đổi chuỗi sang số nguyên
50.
51.         Console.ForegroundColor = ConsoleColor.Magenta;
52.         Console.Write("Reading [y/n]: ");
53.         Console.ResetColor();
54.         ConsoleKeyInfo readingChar = Console.ReadKey(); //đọc 1 ký tự và lưu vào biế
55.         bool reading = readingChar.KeyChar == 'y' || readingChar.KeyChar == 'Y' ?
56.             true : false; //chuyển sang kiểu bool dùng biểu thức điều kiện
57.         Console.WriteLine();
58.
59.         // TODO: TẠM DỪNG Ở ĐÂY, SẼ QUAY LẠI SAU
60.     }
61. }
62. }

```

Bước 3. Bổ sung phương thức Create cho BookController

Bổ sung phương thức Create vào cuối class BookController 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.Controllers
8.     {
9.         using Models; //lưu ý cách dùng using với không gian tên con
10.        using Views;
11.
12.        /// <summary>
13.        /// lớp điều khiển, giúp ghép nối dữ liệu sách với giao diện
14.        /// </summary>
15.        internal class BookController
16.        {
17.            /// <summary>
18.            /// ghép nối dữ liệu 1 cuốn sách với giao diện hiển thị 1 cuốn sách
19.            /// </summary>
20.            /// <param name="id">mã định danh của cuốn sách</param>
21.            public void Single(int id)

```

```

22.     {
23.         // khởi tạo object với property
24.         Book model = new Book
25.         {
26.             Id = 1,
27.             Authors = "Adam Freeman",
28.             Title = "Expert ASP.NET Web API 2 for MVC Developers (The Expert's Voice",
29.             Publisher = "Apress",
30.             Year = 2014,
31.             Tags = "c#, asp.net, mvc",
32.             Description = "Expert insight and understanding of how to create, custom",
33.             Rating = 5,
34.             Reading = true
35.         };
36.
37.         // khởi tạo view
38.         BookSingleView view = new BookSingleView(model);
39.         // gọi phương thức Render để thực sự hiển thị ra màn hình
40.         view.Render();
41.     }
42.
43.     /// <summary>
44.     /// kích hoạt chức năng nhập dữ liệu cho 1 cuốn sách
45.     /// </summary>
46.     public void Create()
47.     {
48.         BookCreateView view = new BookCreateView(); // khởi tạo object
49.         view.Render(); // hiển thị ra màn hình
50.     }
51. }
52. }

```

Để nhập dữ liệu từ giao diện dòng lệnh có thể sử dụng các phương thức của lớp `Console`, bao gồm `ReadLine`, `ReadKey`, `Read`. Trong đó:

- `ReadLine` đọc một dòng và trả về một chuỗi ký tự.
- `ReadKey` đọc một ký tự và trả về kiểu `ConsoleKeyInfo`.
- `Read` đọc một ký tự và trả về mã `ascii` của ký tự đó.

Thực thi lệnh theo yêu cầu của người dùng

Bạn có thể nhận thấy một vấn đề, mặc dù chúng ta đã chế tạo lớp view thứ hai và phương thức của controller tương ứng nhưng lại chưa thể thực thi lệnh theo yêu cầu của người dùng.

Ở bài này các bài tiếp theo, chúng ta sẽ xây dựng thêm các lớp view mới để đáp ứng hết các nhu cầu tương tác với người dùng (như hiển thị danh sách, nhập dữ liệu mới, cập nhật dữ liệu, xóa dữ liệu).

Ứng với mỗi lớp view, theo quy ước của MVC chúng ta đang áp dụng, sẽ có một phương thức tương ứng của controller làm nhiệm vụ ghép nối giữa dữ liệu (model) với giao diện (view).

Như vậy, chương trình cần có khả năng tiếp nhận yêu cầu của người dùng và kích hoạt phương thức tương ứng của controller.

Trong các MVC framework (như ASP.NET MVC) đều có một công cụ làm nhiệm vụ đó và thường được gọi là *router*. Router cho phép ánh xạ một yêu cầu của người dùng (ví dụ, dưới

dạng một chuỗi ký tự lệnh cùng tham số trong ứng dụng console, hoặc một chuỗi url đối với ứng dụng web) sang việc thực thi một phương thức tương ứng của controller.

Với kỹ thuật C# hiện tại chúng ta chưa đủ khả năng để tạo ra một router đúng nghĩa. Tuy nhiên, để giải quyết phần nào bài toán, sau đây chúng ta sẽ sử dụng cấu trúc rẽ nhiều nhánh để mô phỏng khả năng của một router.

Thực hành 2: nhận lệnh từ người dùng

Bước 1. Chỉnh sửa code của phương thức Main

```
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
8. {
9.     using Controllers;
10.
11.     internal class Program
12.     {
13.         private static void Main(string[] args)
14.         {
15.             BookController controller = new BookController();
16.
17.             while (true)
18.             {
19.                 Console.Write("Request> ");
20.                 string request = Console.ReadLine();
21.
22.                 switch (request.ToLower())
23.                 {
24.                     case "single":
25.                         controller.Single(1);
26.                         break;
27.
28.                     default:
29.                         Console.WriteLine("Unknown command");
30.                         break;
31.                 }
32.             }
33.         }
34.     }
35. }
```

Bước 2. Dịch và chạy thử chương trình với lệnh single

Gõ lệnh single từ dấu nhắc lệnh

```
D:\_Temp\BOOKMAN\BookMan.ConsoleApp\bin\Debug\BookMan.Consol...
Request> single
BOOK DETAIL INFORMATION
Authors: Adam Freeman
Title: Expert ASP.NET Web API 2 for MVC Developers (The Expert's Voice in .NET)
Publisher: Apress
Year: 2014
Edition: 1
Isbn:
Tags: c#, asp.net, mvc
Description: Expert insight and understanding of how to create, customize, and deploy complex, flexible, and robust HTTP web services
Rating: 5
Reading: True
File:
File Name:
Request> asdf
Unknown command
Request> _
```

Kết quả chạy chương trình

Nếu nhập đúng lệnh `single` (không phân biệt hoa thường), phương thức `Single` của controller sẽ được gọi. Khi nhập bất kỳ lệnh nào khác sẽ viết ra màn hình "Unknown command".

Bước 3. Bổ sung khả năng thực hiện lệnh create

Bổ sung thêm một nhánh cho cấu trúc switch-case của `Main` (lớp `Program`)

```
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
8. {
9.     using Controllers;
10.
11.     internal class Program
12.     {
13.         private static void Main(string[] args)
14.         {
15.             BookController controller = new BookController();
16.
17.             while (true)
18.             {
19.                 Console.Write("Request> ");
20.                 string request = Console.ReadLine();
21.
22.                 switch (request.ToLower())
23.                 {
24.                     case "single":
25.                         controller.Single(1);
26.                         break;
27.
28.                     case "create":
29.                         controller.Create();
30.                         break;
31.
32.                     default:
33.                         Console.WriteLine("Unknown command");
34.                         break;
35.                 }
36.             }
37.         }
38.     }
39. }
```

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

```
D:\Temp\BOOKMAN\BookMan.ConsoleApp\bin\Debug\BookMan.Consol...
Request> create
CREATE A NEW BOOK
Title: C# programming for dummy
Authors: Donald Trump
Publisher: The White House
Year: 2018
Edition: 1
Reading [y/n]: y
Request> _
```

Kết quả chạy chương trình với lệnh create

Trong phương thức `Main` ở trên, chúng ta chuyển đổi chuỗi truy vấn của người dùng thành dạng chữ thường (để người dùng không phải quan tâm nhập chữ thường/chữ hoa) bằng phương thức `ToLower` của lớp `string` và cung cấp biểu thức này cho cấu trúc switch.

Trong cấu trúc này chúng ta mới xác định được một "case": nếu giá trị chuỗi truy vấn (sau khi chuyển về chữ thường) là "single" thì sẽ gọi phương thức `Single` của controller. Trong các bài tiếp theo chúng ta sẽ lần lượt bổ sung thêm các "case" mới. Nếu giá trị của chuỗi truy vấn không trùng với bất kỳ "case" nào, chúng ta sẽ in ra màn hình thông báo "Unknown command".

Thực hành 3: cải tiến BookCreateView

Bước 1. Bổ sung phương thức in ra console với màu sắc

Bổ sung thêm hai phương thức sau vào cuối lớp `BookCreateView`

```
1.  /// <summary>
2.  /// xuất thông tin ra console với màu sắc (WriteLine có màu)
3.  /// </summary>
4.  /// <param name="message">thông tin cần xuất</param>
5.  /// <param name="color">màu chữ</param>
6.  /// <param name="resetColor">trả lại màu mặc định hay không</param>
7.  private void WriteLine(string message, ConsoleColor color = ConsoleColor.White, bool res
8.  {
9.      Console.ForegroundColor = color;
10.     Console.WriteLine(message);
11.     if (resetColor)
12.         Console.ResetColor();
13. }
14.
15.  /// <summary>
16.  /// xuất thông tin ra console với màu sắc (Write có màu)
17.  /// </summary>
18.  /// <param name="message">thông tin cần xuất</param>
19.  /// <param name="color">màu chữ</param>
20.  /// <param name="resetColor">trả lại màu mặc định hay không</param>
21.  private void Write(string message, ConsoleColor color = ConsoleColor.White, bool resetCo
22.  {
23.      Console.ForegroundColor = color;
24.      Console.Write(message);
25.      if (resetColor)
26.          Console.ResetColor();
27. }
```

Bước 2. Sửa code của phương thức Render

```
1.  public void Render()
2.  {
3.      WriteLine("CREATE A NEW BOOK", ConsoleColor.Green);
4.  }
```

```

5.    ConsoleColor labelColor = ConsoleColor.Magenta;
6.
7.    Write("Title: ", labelColor);
8.    var title = Console.ReadLine(); //đọc 1 dòng và lưu vào biến title
9.
10.   Write("Authors: ", labelColor);
11.   var authors = Console.ReadLine(); //đọc 1 dòng và lưu vào biến authors
12.
13.   Write("Publisher: ", labelColor);
14.   var publisher = Console.ReadLine(); //đọc 1 dòng và lưu vào biến publisher
15.
16.   Write("Year: ", labelColor);
17.   string yearString = Console.ReadLine(); //đọc 1 dòng và lưu vào biến yearString
18.   var year = int.Parse(yearString); //chuyển đổi chuỗi sang số nguyên
19.
20.   Write("Edition: ", labelColor);
21.   var editionString = Console.ReadLine(); //đọc 1 dòng và lưu vào biến editionString
22.   var edition = int.Parse(editionString); //chuyển đổi chuỗi sang số nguyên
23.
24.   Write("Reading [y/n]: ", labelColor);
25.   ConsoleKeyInfo readingChar = Console.ReadKey(); //đọc 1 ký tự và lưu vào biến yearSt
26.   var reading = readingChar.KeyChar == 'y' || readingChar.KeyChar == 'Y' ?
27.       true : false; //chuyển sang kiểu bool dùng biểu thức điều kiện
28.   Console.WriteLine();
29.
30.   Write("Tags: ", labelColor);
31.   var tags = Console.ReadLine();
32.
33.   Write("Description: ", labelColor);
34.   var description = Console.ReadLine();
35.
36.   Write("Rate: ", labelColor);
37.   var rateString = Console.ReadLine();
38.   var rating = int.Parse(rateString);
39.
40.   Write("File: ", labelColor);
41.   var file = Console.ReadLine();
42. }

```

Trong lần cải tiến này, chúng ta viết thêm hai phương thức giúp in chữ có màu ra console và dùng hai phương thức này để giảm bớt số lượng code bị lặp trong phương thức Render. Chúng ta cũng bổ sung code để nhập nốt các dữ liệu còn lại của sách.

Trong lần cải tiến trên chúng ta xây dựng hai phương thức mới Write và WriteLine với danh sách tham số có chút khác biệt với những phương thức bình thường. Trong hai phương thức này, tham số `color` và `resetColor` được gán sẵn giá trị: `color = ConsoleColor.White`, và `resetColor = true`.


Tính năng này của C# được gọi là *tham số với giá trị mặc định* hoặc *tham số không bắt buộc* hoặc *tham số tùy chọn* (Optional Arguments / Parameters).

Các phương thức Write và WriteLine ở trên mặc dù được định nghĩa với 3 tham số vào nhưng hai tham số sau là tham số tùy chọn: một tham số nhận màu sắc mặc định là White; một tham số nhận giá trị mặc định là true.

Do đó, khi gọi các phương thức này trong `Render`, chúng ta chỉ cung cấp giá trị cho tham số `color` (do chúng ta muốn viết ra chữ màu Magenta, khác với màu White mặc định) nhưng không cung cấp giá trị cho tham số thứ 3 (vì vẫn muốn dùng giá trị true, vốn là giá trị có sẵn của tham số tùy chọn này).

```
1 reference
private void WriteLine(string message, ConsoleColor color = ConsoleColor.White, bool resetColor = true)
{
    Console.ForegroundColor = color;
    Console.WriteLine(message);
    if (resetColor)
        Console.ResetColor();
}

```

 void BookCreateView.WriteLine(string message, [ConsoleColor color = ConsoleColor.White], [bool resetColor = true])
xuất thông tin ra console với màu sắc (WriteLine có màu)

Cách Visual Studio hiển thị thông tin hỗ trợ của tham số tùy chọn.

Thực hành 4: tiếp tục cải tiến BookCreateView

Trong phần thực hành này chúng ta lại tiếp tục xây dựng thêm các phương thức mới giúp đơn giản hóa hơn nữa việc nhập dữ liệu, đồng thời luyện tập cách xây dựng phương thức.

Bước 1. Xây dựng phương thức nhập dữ liệu từ Console

Bổ sung các phương thức sau vào lớp `BookCreateView` : `InputBool` , `InputInt` , `InputString`

```
1.  /// <summary>
2.  /// in ra thông báo và tiếp nhận chuỗi ký tự người dùng nhập
3.  /// rồi chuyển sang kiểu bool
4.  /// </summary>
5.  /// <param name="label">dòng thông báo</param>
6.  /// <param name="labelColor">màu chữ thông báo</param>
7.  /// <param name="valueColor">màu chữ người dùng nhập</param>
8.  /// <returns></returns>
9.  private bool InputBool(string label, ConsoleColor labelColor = ConsoleColor.Magenta, Con
10. {
11.     Write($"{label} [y/n]: ", labelColor);
12.     ConsoleKeyInfo key = Console.ReadKey(); //đọc 1 ký tự vào biến key
13.     Console.WriteLine();
14.     bool @char = key.KeyChar == 'y' || key.KeyChar == 'Y' ?
15.         true : false; //chuyển sang kiểu bool dùng biểu thức điều kiện
16.     return @char; // lưu ý cách viết tên biến @char
17. }
18.
19. /// <summary>
20. /// in ra thông báo và tiếp nhận chuỗi ký tự người dùng nhập
21. /// rồi chuyển sang số nguyên
22. /// </summary>
23. /// <param name="label">dòng thông báo</param>
24. /// <param name="labelColor">màu chữ thông báo</param>
25. /// <param name="valueColor">màu chữ người dùng nhập</param>
26. /// <returns></returns>
27. private int InputInt(string label, ConsoleColor labelColor = ConsoleColor.Magenta, Conso
28. {
29.     while (true)
30.     {
31.         var str = InputString(label, labelColor, valueColor);
32.         var result = int.TryParse(str, out int i);
33.         if (result == true)
34.         {
35.             return i;
36.         }
37.     }
38. }
39.
40. /// <summary>
41. /// in ra thông báo và tiếp nhận chuỗi ký tự người dùng nhập
42. /// </summary>
43. /// <param name="label">dòng thông báo</param>
44. /// <param name="labelColor">màu chữ thông báo</param>
45. /// <param name="valueColor">màu chữ người dùng nhập</param>
46. /// <returns></returns>
47. private string InputString(string label, ConsoleColor labelColor = ConsoleColor.Magenta,
48. {
49.     Write($"{label}: ", labelColor, false);
50.     Console.ForegroundColor = valueColor;
51.     string value = Console.ReadLine();
52.     Console.ResetColor();

```



```
53.         return value;
54.     }

```

Bước 2. Cải tiến code của phương thức Render

Sử dụng 3 phương thức mới để làm đơn giản code của phương thức Render

```
1. public void Render()
2. {
3.     WriteLine("CREATE A NEW BOOK", ConsoleColor.Green);
4.
5.     var title = InputString("Title"); //đọc vào biến title
6.     var authors = InputString("Authors"); //đọc vào biến authors
7.     var publisher = InputString("Publisher"); //đọc vào biến publisher
8.     var year = InputInt("Year"); // nhập giá trị cho biến year
9.     var edition = InputInt("Edition"); // nhập giá trị cho biến edition
10.    var tags = InputString("Tags");
11.    var description = InputString("Description");
12.    var rate = InputInt("Rate");
13.    var reading = InputBool("Reading");
14.    var file = InputString("File");
15. }
```

Code đầy đủ của lớp BookCreateView như sau:

```
1. using System;
2.
3. namespace BookMan.ConsoleApp.Views
4. {
5.     /// <summary>
6.     /// class để thêm một cuốn sách mới
7.     /// </summary>
8.     internal class BookCreateView
9.     {
10.         public BookCreateView()
11.         {
12.
13.         }
14.
15.         /// <summary>
16.         /// yêu cầu người dùng nhập từng thông tin và lưu lại thông tin đó
17.         /// </summary>
18.         public void Render()
19.         {
20.             WriteLine("CREATE A NEW BOOK", ConsoleColor.Green);
21.
22.             var title = InputString("Title"); //đọc vào biến title
23.             var authors = InputString("Authors"); //đọc vào biến authors
24.             var publisher = InputString("Publisher"); //đọc vào biến publisher
25.             var year = InputInt("Year"); // nhập giá trị cho biến year
26.             var edition = InputInt("Edition"); // nhập giá trị cho biến edition
27.             var tags = InputString("Tags");
28.             var description = InputString("Description");
29.             var rate = InputInt("Rate");
30.             var reading = InputBool("Reading");
31.             var file = InputString("File");
32.         }
33.
34.
35.         /// <summary>
36.         /// xuất thông tin ra console với màu sắc (WriteLine có màu)
37.         /// </summary>
38.         /// <param name="message">thông tin cần xuất</param>
39.         /// <param name="color">màu chữ</param>
40.         /// <param name="resetColor">trả lại màu mặc định hay không</param>
41.         private void WriteLine(string message, ConsoleColor color = ConsoleColor.White,
42.         {
43.             Console.ForegroundColor = color;
44.             Console.WriteLine(message);
45.             if (resetColor)
46.                 Console.ResetColor();
47.         }
48.
49.         /// <summary>

```

```

50.    /// xuất thông tin ra console với màu sắc (Write có màu)
51.    /// </summary>
52.    /// <param name="message">thông tin cần xuất</param>
53.    /// <param name="color">màu chữ</param>
54.    /// <param name="resetColor">trả lại màu mặc định hay không</param>
55.    private void Write(string message, ConsoleColor color = ConsoleColor.White, bool
56.    {
57.        Console.ForegroundColor = color;
58.        Console.Write(message);
59.        if (resetColor)
60.            Console.ResetColor();
61.    }
62.
63.    /// <summary>
64.    /// in ra thông báo và tiếp nhận chuỗi ký tự người dùng nhập
65.    /// rồi chuyển sang kiểu bool
66.    /// </summary>
67.    /// <param name="label">dòng thông báo</param>
68.    /// <param name="labelColor">màu chữ thông báo</param>
69.    /// <param name="valueColor">màu chữ người dùng nhập</param>
70.    /// <returns></returns>
71.    private bool InputBool(string label, ConsoleColor labelColor = ConsoleColor.Mage
72.    {
73.        Write($"{label} [y/n]: ", labelColor);
74.        ConsoleKeyInfo key = Console.ReadKey(); //đọc 1 ký tự vào biến key
75.        Console.WriteLine();
76.        bool @char = key.KeyChar == 'y' || key.KeyChar == 'Y' ?
77.            true : false; //chuyển sang kiểu bool dùng biểu thức điều kiện
78.        return @char; // lưu ý cách viết tên biến @char
79.    }
80.
81.    /// <summary>
82.    /// in ra thông báo và tiếp nhận chuỗi ký tự người dùng nhập
83.    /// rồi chuyển sang số nguyên
84.    /// </summary>
85.    /// <param name="label">dòng thông báo</param>
86.    /// <param name="labelColor">màu chữ thông báo</param>
87.    /// <param name="valueColor">màu chữ người dùng nhập</param>
88.    /// <returns></returns>
89.    private int InputInt(string label, ConsoleColor labelColor = ConsoleColor.Magenta
90.    {
91.        while (true)
92.        {
93.            var str = InputString(label, labelColor, valueColor);
94.            var result = int.TryParse(str, out int i);
95.            if (result == true)
96.            {
97.                return i;
98.            }
99.        }
100.    }
101.
102.    /// <summary>
103.    /// in ra thông báo và tiếp nhận chuỗi ký tự người dùng nhập
104.    /// </summary>
105.    /// <param name="label">dòng thông báo</param>
106.    /// <param name="labelColor">màu chữ thông báo</param>
107.    /// <param name="valueColor">màu chữ người dùng nhập</param>
108.    /// <returns></returns>
109.    private string InputString(string label, ConsoleColor labelColor = ConsoleColor.
110.    {
111.        Write($"{label}: ", labelColor, false);
112.        Console.ForegroundColor = valueColor;
113.        string value = Console.ReadLine();
114.        Console.ResetColor();
115.        return value;
116.    }
117.
118.
119.    }
120.    }

```

Bước 3. Dịch và chạy thử chương trình

Chạy thử lệnh create như sau:

Kết quả chạy chương trình

Để ý trong phương thức InputBool ở trên chúng ta sử dụng **phép toán điều kiện** (conditional operator).

```
1. ConsoleKeyInfo key = Console.ReadKey(); //đọc 1 ký tự vào biến key
2. bool @char = key.KeyChar == 'y' || key.KeyChar == 'Y' ?
3.     true : false; //chuyển sang kiểu bool dùng biểu thức điều kiện
```

Kết luận

Chúng ta đã xây dựng được một class cho phép nhập dữ liệu một cuốn sách từ giao diện console. Qua bài này chúng ta đã học cách vận dụng **biến cục bộ**, nhập dữ liệu từ **console**, **cấu trúc điều khiển**, **biến đổi kiểu**, **tham số của phương thức** (tham số tùy chọn, tham số out).

- + 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!