Name: Lý Định Công Minh

Student Code: SE140952

Class: SE05D

**Lab 1 – Book Management**

Code hàm mainBookManagement:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using BookManagement;

namespace mainBookManagement

{

class Program

{

public static int menu()

{

Console.WriteLine("\n\*\*\*\*\*\*Book Management\*\*\*\*\*\*");

Console.WriteLine("1. Add Book");

Console.WriteLine("2. Find book");

Console.WriteLine("3. Show Book");

Console.WriteLine("4. Remove Book");

Console.WriteLine("5. Exit Book");

int choice = 0;

Console.Write("Input Command: ");

choice = int.Parse(Console.ReadLine());

return choice;

}

public static void Main(string[] args)

{

int choice;

BookLib book = new BookLib();

do

{

choice = menu();

switch (choice)

{

case 1:

{

String move;

do

{

book.GetBook();

Console.WriteLine("\nDo you want to continue? (Yes/No)");

move = Console.ReadLine();

} while (move.Equals("Yes"));

break;

}

case 2:

{

String move;

do

{

book.SearchBook();

Console.WriteLine("\nDo you want to continue? (Yes/No)");

move = Console.ReadLine();

} while (move.Equals("Yes"));

break;

}

case 3:

{

String move;

do

{

book.Show();

Console.WriteLine("\nDo you want to continue? (Yes/No)");

move = Console.ReadLine();

} while (move.Equals("Yes"));

break;

}

case 4:

{

String move;

do

{

book.RemoveBook();

Console.WriteLine("\nDo you want to continue? (Yes/No)");

move = Console.ReadLine();

} while (move.Equals("Yes"));

break;

}

case 5:

{

Console.WriteLine("Thank you !!!");

break;

}

}

if (choice > 5) {

Console.WriteLine("Please input command from 1 to 5 !!!");

}

} while (choice >= 1 && choice <= 4);

}

}

}

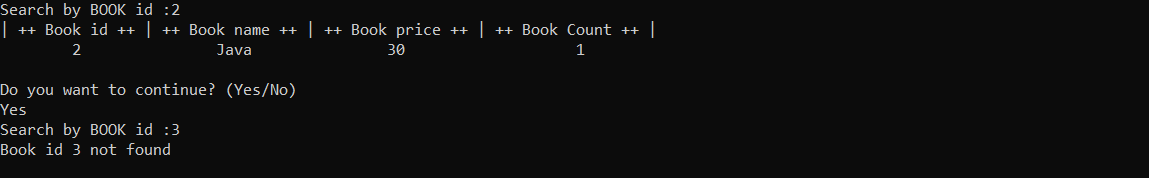
* Màn hình console khi ta chạy chương trình:



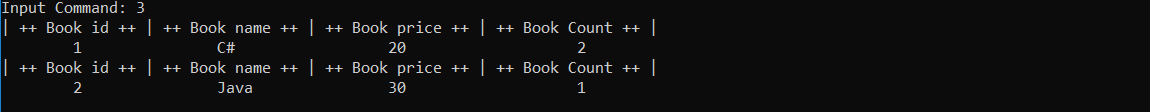
* Người dùng sẽ thực hiện chọn 1 trong các tác vụ trên, đầu tiên người dùng chọn function 1. Add Book. Khi người dùng chọn 1, chương trình sẽ tạo ID Book đầu tiên với giá trị 1, và tăng dần sau mỗi lần Add ( để tránh tình trạng bị trùng lặp ID Book ). Người dùng sẽ nhập thêm thông tin gồm tên sách, giá sách, và số lượng của sách. Sau khi điền đẩy đủ các trường dữ liệu, chương trình sẽ in ra câu thông báo để xác nhận với người dùng rằng có muốn tiếp tục ở chức năng này hay không. Nếu người dùng chọn “ Yes “ thì hàm sẽ tiếp tục chạy, và có thể Add thêm nhiều sách khác. Nếu người dùng chọn “ No “ thì sẽ hiện ra lại menu để người dùng tiếp tục thao tác



* Nếu người dùng chọn chức năng 2. Find Book. Chương trình sẽ in ra màn hình yêu cầu người dùng nhập ID Book, nếu ID đó có tồn tại, chương trình sẽ trả về kết quả các thuộc tính của ID đó. Nếu ID không tồn tại, chương trình sẽ in ra thông báo, “ Book id not found “



* Nếu người dùng chọn chức năng 3.Show Book, chương trình sẽ in ra 1 danh sách tất cả các dữ liệu chứa trong List và in tất cả ra màn hình



* Nếu người dùng chọn chức năng 4.Remove Book, chương trình sẽ yêu cầu người dùng nhập vào ID Book, nếu ID đó có tồn tại, sẽ thực hiện hàm xoá ID đó ra khỏi List. Trường hợp nếu ID đó không tồn tại, sẽ in ra câu thông báo “ Invalid Book id “

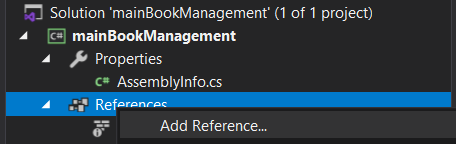


* Chức năng cuối cùng 5.Exit là câu lệnh in ra màn hình “ Thank you!!! ” và dừng chương trình

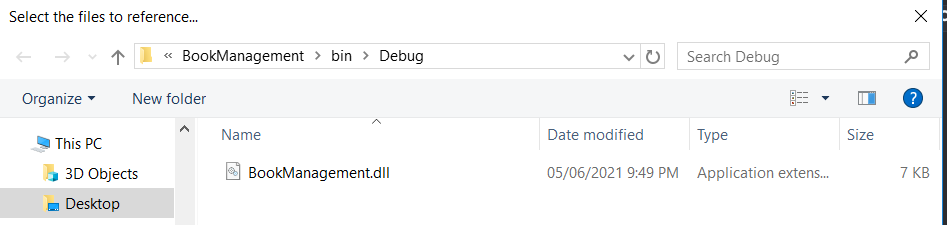


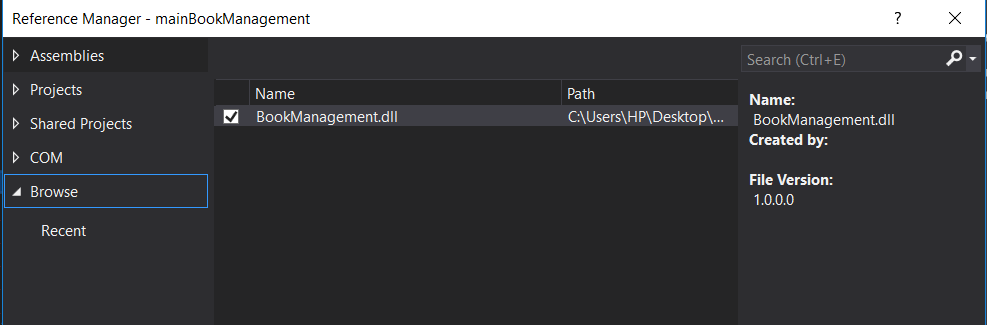
* Để hiện thực được chương trình trên ta cần import file dll, gồm các bước như sau:

+ Đầu tiền ở khung Solution -> Reference -> Add Reference



+ Tiếp theo Add file dll theo đường dẫn: C:\Users\HP\Desktop\PRN292\SE140952\_BookManagement\BookManagement\BookManagement\bin\Debug





+ Tiếp theo ta sử dụng file dll bằng import như sau, và khởi tạo Object để truy xuất đến các hàm trong file dll





Code file dll BookManagement:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BookManagement

{

public class BookLib

{

class Book

{

public int bookId;

public string bookName;

public int bookPrice;

public int bookCount;

public int x;

}

static List<Book> bookList = new List<Book>();

static Book book = new Book();

public void GetBook()

{

String price;

Book book = new Book();

Console.WriteLine("\nBook Id:{0}", book.bookId = bookList.Count + 1);

Console.Write("Book Name:");

book.bookName = Console.ReadLine();

do

{

Console.Write("Book Price:");

price = Console.ReadLine();

} while (int.TryParse(price, out book.bookPrice) == false);

Console.Write("Number of Books:");

book.x = book.bookCount = int.Parse(Console.ReadLine());

bookList.Add(book);

}

public void RemoveBook()

{

Book book = new Book();

Console.Write("Enter Book id to be deleted : ");

int Del = int.Parse(Console.ReadLine());

if (bookList.Exists(x => x.bookId == Del))

{

bookList.RemoveAt(Del - 1);

Console.WriteLine("Book id - {0} has been deleted", Del);

}

else

{

Console.WriteLine("Invalid Book id");

}

bookList.Add(book);

}

public void SearchBook()

{

Book book = new Book();

Console.Write("Search by BOOK id :");

int find = int.Parse(Console.ReadLine());

if (bookList.Exists(x => x.bookId == find))

{

foreach (Book searchId in bookList)

{

if (searchId.bookId == find)

{

Console.WriteLine("| ++ Book id ++ |" + " ++ Book name ++ |" + " ++ Book price ++ |" + " ++ Book Count ++ |");

Console.WriteLine(("\t" + searchId.bookId + " ") + ("\t\t" + searchId.bookName + " ") + ("\t\t " + searchId.bookPrice + " ") + ("\t\t\t" + searchId.bookCount + " "));

/\*Console.WriteLine("Book id :{0}\n" +

"Book name :{1}\n" +

"Book price :{2}\n" +

"Book Count :{3}", searchId.bookId, searchId.bookName, searchId.bookPrice, searchId.bookCount);\*/

}

}

}

else

{

Console.WriteLine("Book id {0} not found", find);

}

}

public void Show()

{

if (book == null)

{

Console.WriteLine("Book not exist!!!");

}

foreach (var book in bookList)

{

Console.WriteLine( "| ++ Book id ++ |" + " ++ Book name ++ |" + " ++ Book price ++ |" + " ++ Book Count ++ |");

Console.WriteLine (("\t" + book.bookId + " ") + ("\t\t" + book.bookName + " ") + ("\t\t " + book.bookPrice + " ") + ("\t\t\t" + book.bookCount + " "));

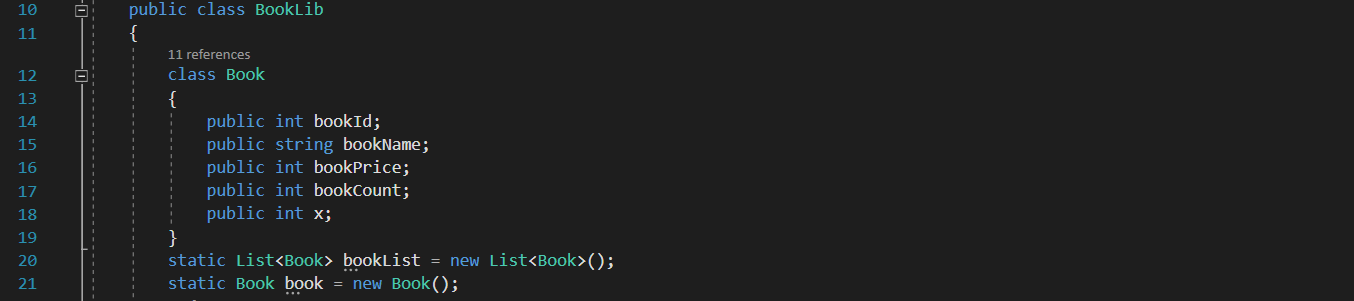
}

}

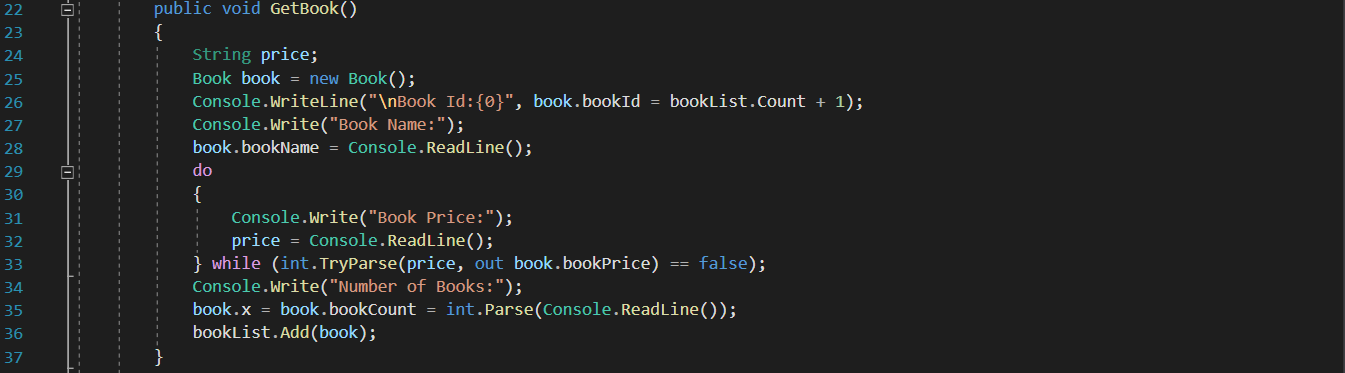
}

}

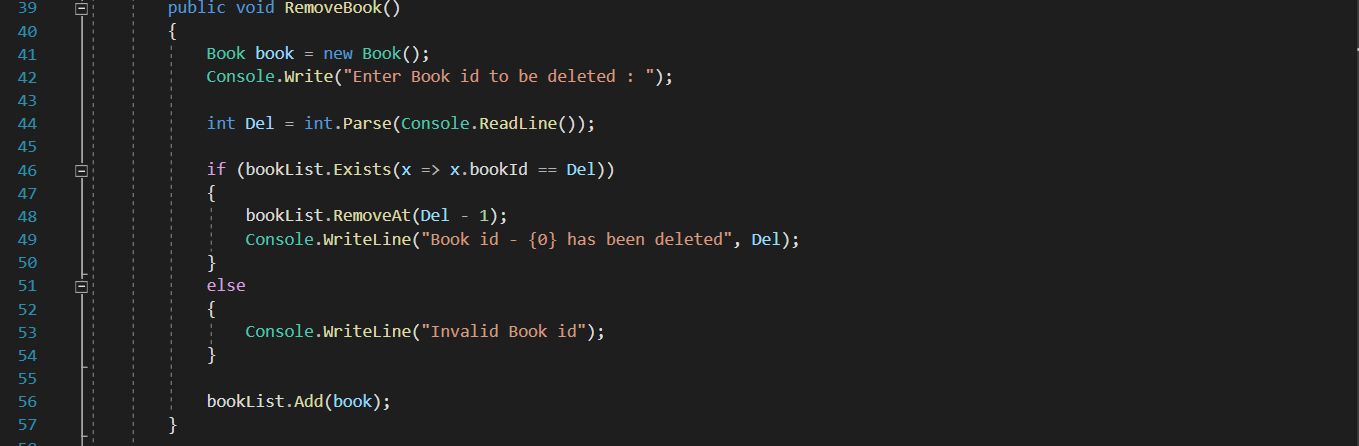
* Đầu tiên ta khởi tạo một class Book chứa các thuộc tính gồm bookID, bookName, bookPrice, bookCount, x. Và một List<Book> để chứa danh sách các phần tử được thêm vào



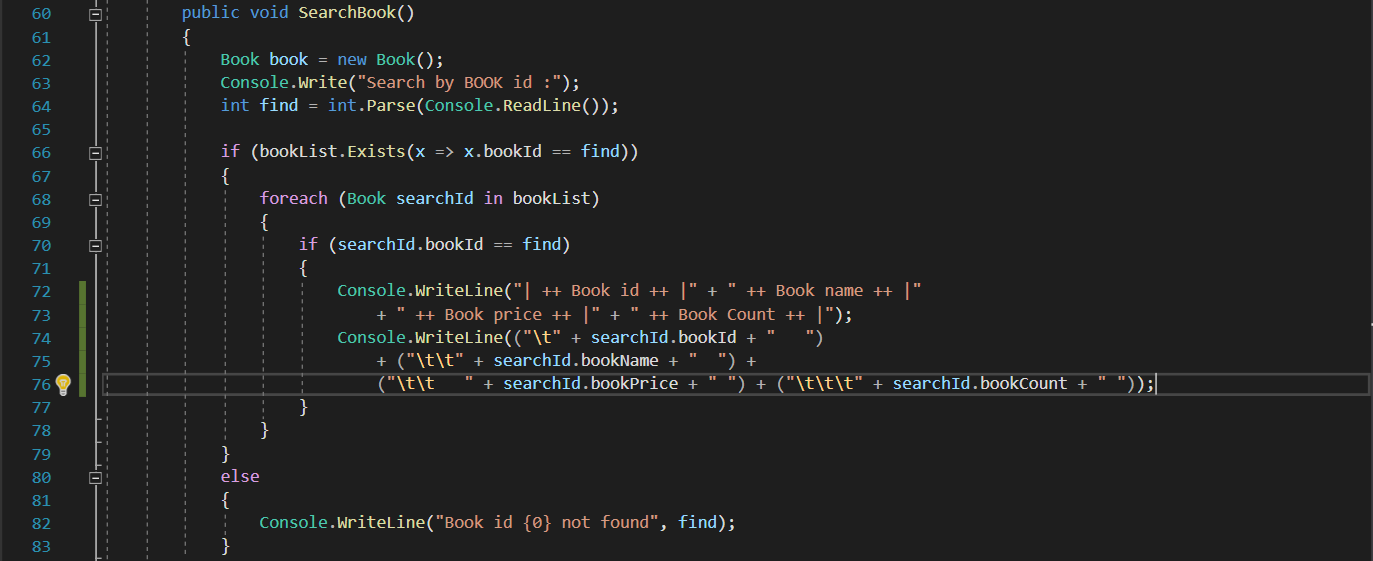
* Tạo một hàm void GetBook() không có kiểu dữ liệu trả về, ban đầu khởi tạo một đối tượng tên book thuộc class Book, tạo ID mặc định ban đầu là 1, và cứ tăng dần nếu người dùng Add thêm. Sau đó cho người dùng nhập vào các thuộc tính của Book gồm Name, Price, Numbers of book. Và sử dụng hàm bookList.Add(book); để thêm dữ liệu vào List.



* Tạo một hàm void RemoveBook() không có kiểu dữ liệu trả về, khởi tạo một đối tượng book tương tự như trên, cho người dùng nhập vào ID Book, cho một biến x chạy và check List bookList, nếu x.bookID == del ( del là ID người dùng nhập ) thì sẽ chạy hàm bookList.RemoveAt(Del – 1) để xoá ID Book đó. Ngược lại nếu biến del người dùng nhập mà biến x không quét được nằm trong bookList, thì sẽ in ra thông báo Invalid Book ID



* Tạo một hàm void SearchBook() để in ra màn hình thông tin các thuộc tính của Book ID. Tạo một biến find cho người dùng nhập vào ID Book, dùng hàm bookList.Exists nếu ID có tồn tại thì dùng vòng lặp foreach in ra thông tin ID Book đó. Trường hợp không tìm thấy ID, sẽ in ra thông báo “ Book id not found “



* Tạo một hàm void Show(), dùng vòng lặp foreach để quét tất cả các phần tử trong bookList để in ra tất cả các phần tử mà không cần thông qua ID Book

