

Propertie, Indexer and Namespace

Bài 1:

Step 1: Tạo giao diện với tên là IBook gồm các thành phần sau:

Properties:

- string ID
- string Name
- string AuthorName
- string Subject
- double BuyingPrice // giá mua vào
- double SellingPrice // giá bán ra

Methods:

- void ShowDetail(),
- void SetDetail(string name, string authorname, string subject, double buyingprice)
- void SetSellingPrice(double newSellingPrice);

Step 2: Xây dựng lớp Book cài đặt giao diện IBook theo yêu cầu sau:

Fields:

- Private static int _Count = 0 // lưu trữ số thể hiện của lớp Book
- Private string _ID;
- Private string _Name
- Private string _AuthorName
- Private _Subject
- Private double _BuyingPrice
- Private double _SellingPrice

Methods:

- **ShowDetail()**, hiển thị thông tin chi tiết về sách
- SetDetail(string name, string authorname, string subject, int buyingPrice), thiết lập thông tin cho sách // giá Bán mặc định sẽ bằng giá mua + 30%
- void SetSellingPrice(double newSellingPrice), thay đổi giá bán ra

Constructors:

Public Book(), khởi tạo giá trị mặc định cho các trường. Trường _ID có giá trị theo cấu trúc B001, B002.....B011, B012B102

Count ++;

ID = "B00" + Count or "B0" + Count hoặc "B" + Count

Properties:

Cài đặt các properties của IBook để thao tác với các trường cùng tên trong lớp Book

Step 3: Xây dựng hàm Main theo yêu cầu sau:

- Tạo ra một thể hiện của Lớp Book với tên theBook
- Nhập thông tin cho một quyển sách (name, authorname, subject, buyingprice)
- theBook.SetDetail(name, authorname, subject, buyingprice);
- Gọi hàm ShowDetail của theBook thông qua tham chiếu của IBook

Bài 2: (sử dụng kết quả trong Step 1, Step 2 của Bài 1)

Step1: Xây dựng giao diện IStorage gồm các thành phần sau:

Properties:

- Int Count

Indexers:

- Book this[int index]

Methods

- Void AddANewBook()
- Void RemoveABook(string id)
- Bool IsBook(string id)

Step 2: Xây dựng lớp **Storage** và cài đặt giao diện **IStorage** theo yêu cầu sau:

Fields:

- private int _Count = 0;
- private Book[] BookStore = new Book[100] // tối đa 100 đầu sách

#region IStorage Members

- Indexer

```
Public Book this[int index]{  
    get{ return BookStore[index];}  
    set{ BookStore[index] = value;}  
}
```

- Properties

```
Public int Count{  
    get{return _Count;}  
    set{ _Count = value;}  
}
```

- Methods

Public void AddANewBook(){ // thêm một quyển sách vào BookStore

- o Yêu cầu người dùng nhập thông tin về một quyển sách
- o Thêm một quyển sách vào BookStore
- o Count++;

}

Public void RemoveABook(string id){ // loại bỏ một Book khỏi BookStore

- o Kiểm tra trong BookStore có BookStore[i].ID ==id
- o Nếu tồn tại thì loại bỏ
- o Thông báo

}

public bool IsABook(string id) { // kiểm tra sự tồn tại của sách trong

BookStore

- o Nếu tồn tại trả về true, trái lại trả về false

}

#endregion

Constructors:

- Public Storage()
 - o Count =0;
 - o Khởi tạo BookStore[]

Step 3: Xây dựng **OnlineShopBook** theo yêu cầu sau

Fields:

- Storage BookStorage = new Storage();
- Double profits = 0; // tiền lãi = Tiền bán – Tiền mua

Methods

- **Public ImportBook()**{ // nhập một số lượng sách
 - o Yêu cầu người dùng cho biết số sách sẽ nhập thêm (n)
 - o Nhập thông tin sách thứ i (name, authorname, subject, sellingprice)
 - o BookStorage[i].SetDetail(name,authorname,subject,sellingprice)
- **Public void ShowDetail()**, hiển thị toàn bộ thông tin sách trong BookStore
- **Public void Show()**, hiển thị toàn bộ thông tin về sách (ngoại trừ giá nhập) trong BookStore
- **Public void SellABook()**, mua một quyển sách
 - o Show()
 - o Yêu cầu nhập mã số sách cần mua (id)
 - o Kiểm tra sự tồn tại của sách tương ứng với id đã nhập
 - o Cập nhật lợi nhuận (**profits**)
 - o BookStorage.RemoveABook(id)
- **Public void ChangeSellingPrice()**, thay đổi giá bán của sách
 - o ShowDetail(), hiển thị toàn bộ thông tin về sách
 - o Yêu cầu nhập mã sách cần thay đổi giá bán (id)
 - o Kiểm tra sự tồn tại (IsBook(id))
 - o Nhập giá tiền bán (newSellingPrice)
 - o BookStorage[i].SetSellingPrice(newSellingPrice);

Step 4: Xây dựng hàm Main theo yêu cầu sau:

- o Tạo một thể hiện OnlineBookShop với tên **BookShop**
- o Hiển thị Menu gồm:
 - 1. Admin
 - 2. Custom
 - 3. Exit
- o Nếu người dùng chọn Admin, thì hiển thị menu:
 - 1. Import book to the store
 - 2. Show the store in detail
 - 3. Change a book selling price
 - 4. Show profits
 - 4. Back
- Nếu người dùng chọn Custom, thì hiển thị menu
 - 1. Show all books of store (**Show()**)
 - 2. Buy a book
 - 3. Back

Bài 3: tách các lớp và giao diện trên thành các file với không gian tên sau:

Book.cs

```
namespace OnlineBookShop
{
    namespace BookShop
    {
        Interface IBook.....
        Class Book.....
    }
}
```

Store.cs

```
namespace OnlineBookShop
{
    namespace StoreShop
    {
        Interface IStorage...
        Class Storage .....
        Class OnlineShop..
    }
}
```

```
namespace OnlineBookShop
{
    Class Program{
        Static void Main(){
            }
        }
}
```

Program.cs

Bài 4: Làm thêm bài tập ATM.doc