using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Assign2

{

public class Book

{

public string Title { get; set; }

public string Author { get; set; }

public string ISBN { get; set; }

public bool IsBorrowed { get; set; }

public Book(string title, string author, string isbn)

{

bool isBorrowed = false;

Title=title;

Author=author;

ISBN=isbn;

}

public void Borrow()

{

IsBorrowed = true;

}

public void Return()

{

IsBorrowed = false;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Remoting.Messaging;

using System.Text;

using System.Threading.Tasks;

namespace Assign2

{

public class Borrower

{

public string Name { get; set; }

public string LibraryCardNumber { get; set; }

public List<Book> BorrowedBooks { get; set; }

public Borrower(string name, string lcn)

{

Name=name;

LibraryCardNumber=lcn;

BorrowedBooks= new List<Book>();

}

public void BorrowBook(Book book)

{

if (!book.IsBorrowed)

{

book.Borrow();

BorrowedBooks.Add(book);

}

}

public void ReturnBook(Book book)

{

if (BorrowedBooks.Contains(book))

{

book.Return();

}

}

}

}

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Assign2

{

public class Library

{

public List<Borrower> Borrowers { get; } = new List<Borrower>();

public List<Book> Books { get; } = new List<Book>();

public Library()

{

Borrowers = new List<Borrower>();

Books = new List<Book>();

}

public void AddBook(Book book)

{

Books.Add(book);

}

public void RegisterBorrower(Borrower borrower)

{

Borrowers.Add(borrower);

}

public void BorrowBook(string isbn, string lCardNumber)

{

// Find the first book that matches the ISBN and is not already borrowed

Book book = null;

foreach (var b in Books)

{

if (b.ISBN == isbn && !b.IsBorrowed)

{

book = b;

break;

}

}

// Find the borrower with the given library card number

Borrower borrower = null;

foreach (var br in Borrowers)

{

if (br.LibraryCardNumber ==lcardNumber)

{

borrower = br;

break;

}

}

// If both found, mark book as borrowed and add to borrower's list

if (book != null && borrower != null)

{

book.IsBorrowed = true;

borrower.BorrowedBooks.Add(book);

}

}

public void ReturnBook(string isbn, string lcardNumber)

{

// Find the borrowed book by ISBN

Book book = null;

foreach (var b in Books)

{

if (b.ISBN == isbn && b.IsBorrowed)

{

book = b;

break;

}

}

// Find the borrower by card number

Borrower borrower = null;

foreach (var br in Borrowers)

{

if (br.LibraryCardNumber == lcardNumber)

{

borrower = br;

break;

}

}

// If both found, mark book as returned and remove from borrower's list

if (book != null && borrower != null)

{

book.IsBorrowed = false;

borrower.BorrowedBooks.Remove(book);

}

}

public List<Book> ViewBooks()

{

return Books;

}

public List<Borrower> ViewBorrowers()

{

return Borrowers;

}

}

}