using NUnit.Framework;

using System.Linq;

using Assign2;

namespace TestLibrary

{

[TestFixture]

public class LibraryTest

{

private Library library;

[SetUp]

public void Setup()

{

library = new Library();

}

[Test]

public void TestAddBook\_ToLibrary()

{

var book = new Book("Book1", "Author1", "ISBN51");

library.AddBook(book);

Assert.Contains(book, library.Books);

}

[Test]

public void TestRegisterBorrower\_ToLibrary()

{

var borrower = new Borrower("Yaswanth", "Card1");

library.RegisterBorrower(borrower);

Assert.Contains(borrower, library.Borrowers);

}

[Test]

public void TestBorrowBook()

{

var book = new Book("Book2", "Author2", "ISBN52");

var borrower = new Borrower("Uday", "Card2");

library.AddBook(book);

library.RegisterBorrower(borrower);

library.BorrowBook("ISBN52", "Card2");

Assert.IsTrue(book.IsBorrowed);

Assert.Contains(book, borrower.BorrowedBooks);

}

[Test]

public void TestReturnBook()

{

var book = new Book("Book3", "Author3", "ISBN53");

var borrower = new Borrower("Pradeep", "Card3");

library.AddBook(book);

library.RegisterBorrower(borrower);

borrower.BorrowBook(book);

library.ReturnBook("ISBN53", "Card3");

Assert.IsFalse(book.IsBorrowed);

Assert.IsFalse(borrower.BorrowedBooks.Contains(book));

}

[Test]

public void TestViewBooks()

{

library.AddBook(new Book("Book1", "Author1", "ISBN51"));

library.AddBook(new Book("Book2", "Author2", "ISBN52"));

var books = library.ViewBooks();

Assert.AreEqual(2, books.Count);

}

[Test]

public void TestViewBorrowers()

{

library.RegisterBorrower(new Borrower("U1", "C1"));

library.RegisterBorrower(new Borrower("U2", "C2"));

var borrowers = library.ViewBorrowers();

Assert.AreEqual(2, borrowers.Count);

}

}

}