package library;

// Interface to track borrow/return

public interface Trackable {

void borrowBook();

void returnBook();

boolean isAvailable();

}

package library;

// Book class (called Volume here, but acts like a book)

public class Volume implements Trackable

{

private String title;

private String author;

private String code;

private boolean available;

public Volume(String title, String author, String code)

{

this.title = title;

this.author = author;

this.code = code;

this.available = true; // new books start as available

}

@Override

public void borrowBook() {

if (!available)

{

System.out.println("This book is already borrowed.");

}

Else

{

available = false;

System.out.println("Book borrowed successfully.");

}

}

@Override

public void returnBook()

{

if (available)

{

System.out.println("This book was not borrowed yet.");

}

else

{

available = true;

System.out.println("Book returned successfully.");

}

}

@Override

public boolean isAvailable()

{

return available;

}

public String getTitle() { return title; }

public String getAuthor() { return author; }

public String getCode() { return code; }

@Override

public String toString()

{

return "\"" + title + "\" by " + author + " (Code: " + code + ") - "

+ (available ? "Available" : "Borrowed");

}

}

package library;

import java.util.ArrayList;

import java.util.List;

// Simple Library class to hold multiple books

public class Library

{

private List<Volume> books = new ArrayList<>();

public void addBook(Volume v)

{

books.add(v);

}

// find a book using its code

public Volume findBook(String code)

{

for (Volume v : books) {

if (v.getCode().equalsIgnoreCase(code))

{

return v;

}

}

return null; // if no match

}

public List<Volume> getAllBooks()

{

return books;

}

}

package library;

import java.util.Scanner;

// Main class -> console menu for testing

public class Main

{

public static void main(String[] args)

{

Library lib = new Library();

Scanner sc = new Scanner(System.in);

boolean running = true;

while (running)

{

System.out.println("\n==== Library Menu ====");

System.out.println("1. Add a Book");

System.out.println("2. Show All Books");

System.out.println("3. Borrow a Book");

System.out.println("4. Return a Book");

System.out.println("0. Exit");

System.out.print("Enter choice: ");

int choice;

if (sc.hasNextInt())

{

choice = sc.nextInt();

sc.nextLine(); // consume newline

} else {

System.out.println("Invalid input! Please enter a number.");

sc.nextLine();

continue;

}

switch (choice) {

case 1:

System.out.print("Title: ");

String t = sc.nextLine();

System.out.print("Author: ");

String a = sc.nextLine();

System.out.print("Book Code: ");

String c = sc.nextLine();

lib.addBook(new Volume(t, a, c));

System.out.println("Book added.");

break;

case 2:

if (lib.getAllBooks().isEmpty()) {

System.out.println("No books in the library.");

} else {

for (Volume v : lib.getAllBooks()) {

System.out.println(v);

}

}

break;

case 3:

System.out.print("Enter code to borrow: ");

String bc = sc.nextLine();

Volume bookToBorrow = lib.findBook(bc);

if (bookToBorrow != null) {

bookToBorrow.borrowBook();

} else {

System.out.println("Book not found.");

}

break;

case 4:

System.out.print("Enter code to return: ");

String rc = sc.nextLine();

Volume bookToReturn = lib.findBook(rc);

if (bookToReturn != null) {

bookToReturn.returnBook();

} else {

System.out.println("Book not found.");

}

break;

case 0:

running = false;

System.out.println("Exiting... Bye!");

break;

default:

System.out.println("Invalid choice.");

}

}

sc.close();

}

}