**Project Name- Library Management Application**

**Main Code:  
  
import java.time.LocalDate;**

**import java.time.format.DateTimeFormatter;**

**import java.time.temporal.ChronoUnit;**

**import java.util.ArrayList;**

**import java.util.Scanner;**

**class Book {**

**int id;**

**String title;**

**String author;**

**String publisher;**

**int publishYear;**

**boolean isBorrowed;**

**String borrowerName;**

**LocalDate borrowDate;**

**int borrowPeriod;**

**Book(int id, String title, String author, String publisher, int publishYear) {**

**this.id = id;**

**this.title = title;**

**this.author = author;**

**this.publisher = publisher;**

**this.publishYear = publishYear;**

**this.isBorrowed = false;**

**this.borrowerName = "";**

**this.borrowDate = null;**

**this.borrowPeriod = 0;**

**}**

**}**

**public class Library {**

**static ArrayList<Book> books = new ArrayList<>();**

**static Scanner scanner = new Scanner(System.in);**

**static int bookIdCounter = 1;**

**static final int FINE\_PER\_DAY = 2;**

**public static void main(String[] args) {**

**int choice;**

**do {**

**System.out.println("\nWelcome to Library Management Application");**

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

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

**System.out.println("3. Show Available Books");**

**System.out.println("4. Borrowed Book");**

**System.out.println("5. Return Book");**

**System.out.println("6. Search Books");**

**System.out.println("7. Exit");**

**System.out.print("\nEnter Your Choice: ");**

**choice = scanner.nextInt();**

**scanner.nextLine();**

**switch (choice) {**

**case 1:**

**addBook();**

**break;**

**case 2:**

**showAllBooks();**

**break;**

**case 3:**

**showAvailableBooks();**

**break;**

**case 4:**

**borrowBook();**

**break;**

**case 5:**

**returnBook();**

**break;**

**case 6:**

**searchBooks();**

**break;**

**case 7:**

**System.out.println("Exiting the application. Goodbye!");**

**break;**

**default:**

**System.out.println("Invalid choice! Please try again.");**

**}**

**} while (choice != 7);**

**}**

**static void addBook() {**

**System.out.print("Enter the title of the book: ");**

**String title = scanner.nextLine();**

**System.out.print("Enter the author of the book: ");**

**String author = scanner.nextLine();**

**System.out.print("Enter the publisher of the book: ");**

**String publisher = scanner.nextLine();**

**System.out.print("Enter the publish year of the book: ");**

**int publishYear = scanner.nextInt();**

**books.add(new Book(bookIdCounter++, title, author, publisher, publishYear));**

**System.out.println("Book added successfully!");**

**}**

**static void showAllBooks() {**

**if (books.isEmpty()) {**

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

**return;**

**}**

**System.out.println("\nList of All Books:\n");**

**System.out.printf("%-5s %-20s %-20s %-15s %-15s %-10s %-20s %-15s %-10s%n",**

**"ID", "TITLE", "AUTHOR", "PUBLISHER", "PUBLISH YEAR", "STATUS", "BORROWER", "BORROW DATE", "PERIOD");**

**for (Book book : books) {**

**System.out.printf("%-5d %-20s %-20s %-15s %-15d %-10s %-20s %-15s %-10s%n",**

**book.id, book.title, book.author, book.publisher, book.publishYear,**

**book.isBorrowed ? "Borrowed" : "Available",**

**book.borrowerName.isEmpty() ? "-" : book.borrowerName,**

**book.borrowDate == null ? "-" : book.borrowDate.format(DateTimeFormatter.ofPattern("dd/MM/yyyy")),**

**book.borrowPeriod > 0 ? book.borrowPeriod + " days" : "-");**

**}**

**}**

**static void showAvailableBooks() {**

**boolean available = false;**

**System.out.println("\nList of Available Books:");**

**System.out.printf("%-5s %-20s %-20s %-15s %-15s%n", "ID", "TITLE", "AUTHOR", "PUBLISHER", "PUBLISH YEAR");**

**for (Book book : books) {**

**if (!book.isBorrowed) {**

**System.out.printf("%-5d %-20s %-20s %-15s %-15d%n",**

**book.id, book.title, book.author, book.publisher, book.publishYear);**

**available = true;**

**}**

**}**

**if (!available) {**

**System.out.println("No books are currently available.");**

**}**

**}**

**static void borrowBook() {**

**System.out.print("Enter the ID of the book to borrow: ");**

**int id = scanner.nextInt();**

**scanner.nextLine();**

**for (Book book : books) {**

**if (book.id == id && !book.isBorrowed) {**

**System.out.print("Enter your name: ");**

**String name = scanner.nextLine();**

**System.out.print("Enter borrow date (dd/MM/yyyy): ");**

**String borrowDateString = scanner.nextLine();**

**System.out.print("Enter period (in days): ");**

**int period = scanner.nextInt();**

**scanner.nextLine();**

**LocalDate borrowDate = LocalDate.parse(borrowDateString, DateTimeFormatter.ofPattern("dd/MM/yyyy"));**

**book.isBorrowed = true;**

**book.borrowerName = name;**

**book.borrowDate = borrowDate;**

**book.borrowPeriod = period;**

**System.out.println("The book is  borrowed : " + book.title);**

**return;**

**}**

**}**

**System.out.println("Sorry, the book is either not available or already borrowed.");**

**}**

**static void returnBook() {**

**System.out.print("Enter the ID of the book to return: ");**

**int id = scanner.nextInt();**

**scanner.nextLine();**

**for (Book book : books) {**

**if (book.id == id && book.isBorrowed) {**

**System.out.print("Enter return date (dd/MM/yyyy): ");**

**String returnDateString = scanner.nextLine();**

**LocalDate returnDate = LocalDate.parse(returnDateString, DateTimeFormatter.ofPattern("dd/MM/yyyy"));**

**long daysBetween = ChronoUnit.DAYS.between(book.borrowDate, returnDate);**

**if (daysBetween > book.borrowPeriod) {**

**long overdueDays = daysBetween - book.borrowPeriod;**

**int fine = (int) (overdueDays \* FINE\_PER\_DAY);**

**System.out.println("Late return! Fine: ₹" + fine);**

**} else {**

**System.out.println("Returned on time. No fine.");**

**}**

**book.isBorrowed = false;**

**book.borrowerName = "";**

**book.borrowDate = null;**

**book.borrowPeriod = 0;**

**System.out.println("Book is returned : " + book.title);**

**return;**

**}**

**}**

**System.out.println("The book is either not borrowed or does not exist.");**

**}**

**static void searchBooks() {**

**System.out.print("Enter keyword to search: ");**

**String keyword = scanner.nextLine().toLowerCase();**

**System.out.println("Search Results:");**

**System.out.printf("%-5s %-20s %-20s %-15s %-15s %-10s %-20s %-15s %-10s%n",**

**"ID", "TITLE", "AUTHOR", "PUBLISHER", "PUBLISH YEAR", "STATUS", "BORROWER", "BORROW DATE", "PERIOD");**

**for (Book book : books) {**

**if (book.title.toLowerCase().contains(keyword) ||**

**book.author.toLowerCase().contains(keyword) ||**

**book.publisher.toLowerCase().contains(keyword) ||**

**Integer.toString(book.publishYear).contains(keyword)) {**

**System.out.printf("%-5d %-20s %-20s %-15s %-15d %-10s %-20s %-15s %-10s%n",**

**book.id, book.title, book.author, book.publisher, book.publishYear,**

**book.isBorrowed ? "Borrowed" : "Available",**

**book.borrowerName.isEmpty() ? "-" : book.borrowerName,**

**book.borrowDate == null ? "-" : book.borrowDate.format(DateTimeFormatter.ofPattern("dd/MM/yyyy")),**

**book.borrowPeriod > 0 ? book.borrowPeriod + " days" : "-");**

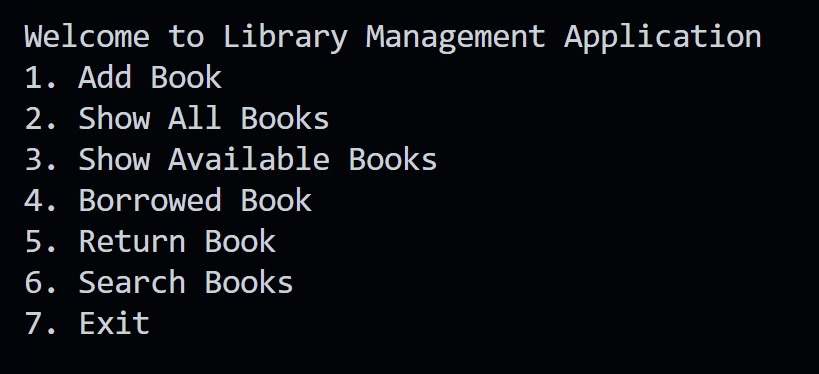
**}**

**}**

**}**

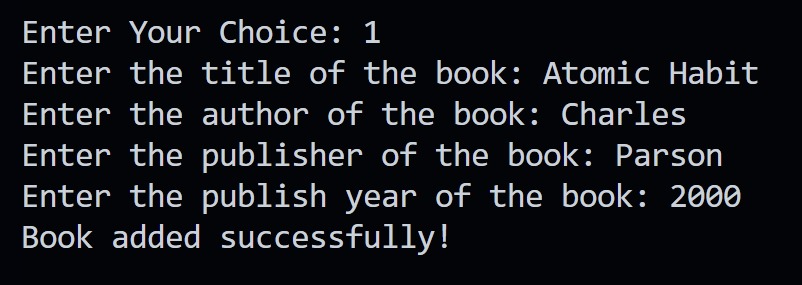
**}**

**Output:**



**Add Book**

* **Feature:** Users can add new books to the library by providing details like:
  + Title of the book.
  + Author's name.
  + Publisher's name.
  + Year of publication.
* **Output:** A confirmation message displays: *"Book added successfully!"* with the book automatically assigned a unique ID.



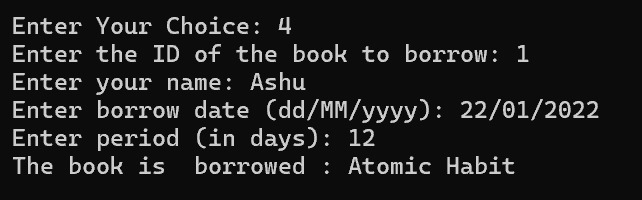
**Show Available Books**

* **Feature:** Lists only the books currently available for borrowing.
* **Output:**
  + Displays a simplified table with:
    - Book ID, Title, Author, Publisher, Year of Publication.
  + If no books are available, it shows: *"No books are currently available."*



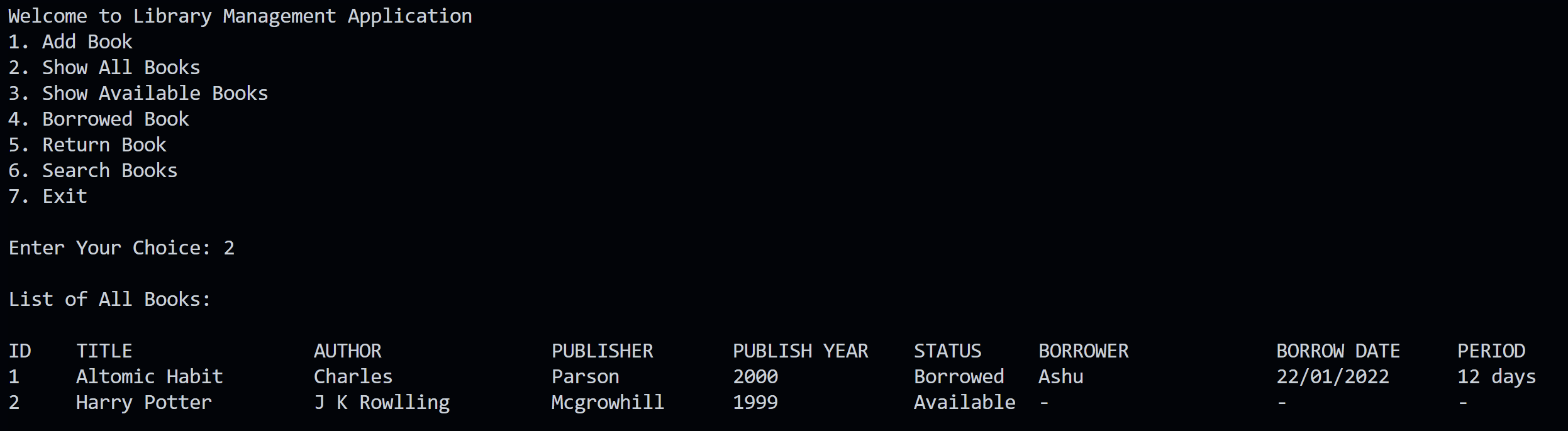
**Borrow a Book**

* **Feature:** Allows a user to borrow a book by providing:
  + Book ID, Borrower's name, Borrow date (in dd/MM/yyyy format), and Borrow period (in days).
* **Output:**
  + If successful: Displays a message like *"The book is borrowed: [Book Title]."*
  + If the book is already borrowed or invalid ID: Displays *"Sorry, the book is either not available or already borrowed."*

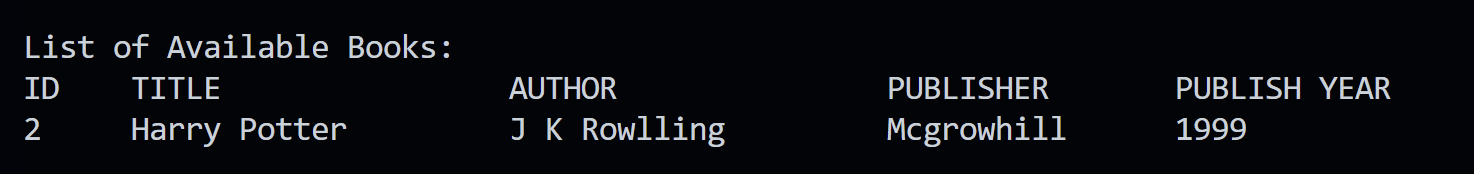


**Show All Books**

* **Feature:** Displays all books in the library, including borrowed and available ones.
* **Output:**
  + Displays a table with detailed information such as:
    - Book ID, Title, Author, Publisher, Year of Publication.
    - Borrow status (Available/Borrowed).
    - Borrower's name (if borrowed).
    - Borrow date and period.



Now, Available Books is changed.



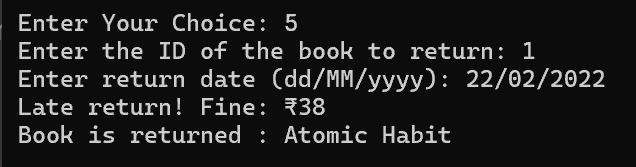
**Search Books**

* **Feature:** Users can search for books using keywords in:
  + Title, Author, Publisher, or Year of Publication.
* **Output:**
  + Displays a table of matching books with all details.
  + If no match is found, the table remains empty.



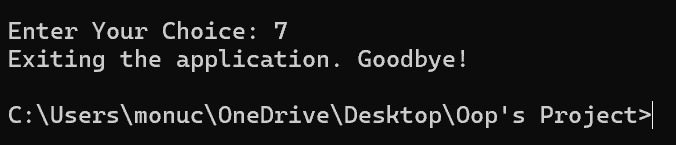
**Return a Book**

* **Feature:** Facilitates returning a borrowed book. Users input:
  + Book ID and the return date.
* **Output:**
  + Calculates the total days the book was held and checks for overdue.
  + If overdue:
    - Calculates the fine based on ₹2 per day.
    - Displays: *"Late return! Fine: ₹[calculated fine amount]."*
  + If returned on time:
    - Displays: *"Returned on time. No fine."*
  + Updates the book’s status to available and resets borrow details.



**Exit**

* **Feature:** Allows users to quit the application.
* **Output:** Displays *"Exiting the application. Goodbye!"*



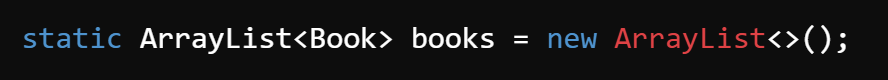
**Error Handling and Additional Functionalities:**

1. **Invalid Choices:** If users enter an invalid menu option, the program displays: *"Invalid choice! Please try again."*
2. **Data Validation:** Ensures valid book IDs for borrowing/returning.
3. **Fine Calculation:** Automatically applies fines for overdue books, making the system realistic and user-friendly.

**The given Library Management Application uses several libraries and frameworks that are part of the Java Standard Library. These support various functionalities like date manipulation, user input handling, and storage. Here's an explanation of the libraries used and their role:**

**1. java.util.ArrayList**

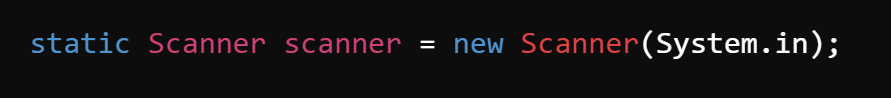
* **Purpose:**
  + This library provides the ArrayList class, a resizable array implementation from the Java Collections Framework.
* **Usage in Code:**
  + It is used to store and manage the list of books in the library dynamically.
  + Example:



* + The ArrayList allows for the efficient addition, retrieval, and updating of books.

**2. java.util.Scanner**

* **Purpose:**
  + The Scanner class is used to read user input from the console.
* **Usage in Code:**
  + It facilitates interactive input for various operations like adding books, borrowing, and returning.
  + Example:



* + This makes the application interactive, allowing the user to enter details like book titles, dates, and IDs.

**3. java.time.LocalDate**

* **Purpose:**
  + The LocalDate class, part of the Java Date-Time API (introduced in Java 8), represents a date without time.
* **Usage in Code:**
  + Used to store and manipulate dates for book borrowing and returning.
  + Example:
    - To parse and format borrow/return dates.
    - To calculate the difference between borrow and return dates for fine calculation.

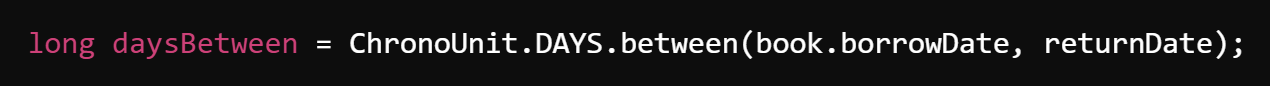
**4. java.time.format.DateTimeFormatter**

* **Purpose:**
  + This class is used to define patterns for formatting and parsing dates.
* **Usage in Code:**
  + To convert user-inputted date strings into LocalDate objects and format LocalDate objects for display.
  + Example Code:



**5. java.time.temporal.ChronoUnit**

* **Purpose:**
  + The ChronoUnit enum provides time units (like days, months, etc.) for calculating differences between dates.
* **Usage in Code:**
  + Used to calculate the number of days between the borrow date and return date.
  + Example Code:



**Summary**

The code extensively leverages Java Standard Library classes, such as ArrayList for data management, Scanner for input handling, and LocalDate for date-related operations. These libraries provide the essential building blocks to create a functional, console-based library management system without relying on external frameworks.

This application is a comprehensive tool for managing a library system. It offers essential operations like adding books, borrowing, returning with fine calculations, and searching for books. It maintains a detailed record of all transactions and ensures an interactive and intuitive user experience.