K.R. Mangalam University

Java programming

Assignment- 4

Submitted by: SOPHIA SONI

Roll No: 2401201042

Course: BCA sp(AI&DS)

Section B

Submitted to: Dr. Manish Kumar

CODE:

Citylibary.java 3 ●

ResultManager.java > Citylibary.java > ...

```java
/*
 * City Library Digital Management System
 * Author: Sophia Soni
 * Email: sophia@gmail.com
 */

import java.io.*;
import java.util.*;

class Book implements Comparable<Book> {
    int bookId;
    String title, author, category;
    boolean isIssued;

    Book(int id, String t, String a, String c) {
        bookId = id;
        title = t;
        author = a;
        category = c;
        isIssued = false;
    }

    public void display() {
        System.out.println(bookId + " | " + title + " | " + author + " | " + category + " | Issued: "
    }

    public int compareTo(Book b) {
        return title.compareToIgnoreCase(b.title);
    }
    public int compareTo(Book b) {
        return title.compareToIgnoreCase(b.title);
    }
}

class Member {
    int memberId;
    String name, email;
    List<Integer> issuedBooks = new ArrayList<>();

    Member(int id, String n, String e) {
        memberId = id;
        name = n;
        email = e;
    }

    public void display() {
        System.out.println(memberId + " | " + name + " | " + email + " | Books: " + issuedBooks);
    }
}

public class Main {
    static Map<Integer, Book> books = new HashMap<>();
    static Map<Integer, Member> members = new HashMap<>();
    static Scanner sc = new Scanner(System.in);
    static int bookId = 101, memberId = 201;

```

```
    Run main | Debug main | Run | Debug
54  public static void main(String[] args) {
55      System.out.println(x: "Welcome to City Library System");
56      while (true) {
57          System.out.println(x: "\n1.Add Book  2.Add Member  3.Issue  4.Return  5.Search  6.Sort
58          int ch = sc.nextInt(); sc.nextLine();
59          switch (ch) {
60              case 1 -> addBook();
61              case 2 -> addMember();
62              case 3 -> issueBook();
63              case 4 -> returnBook();
64              case 5 -> searchBook();
65              case 6 -> sortBooks();
66              case 7 -> { saveData(); System.exit(status: 0); }
67              default -> System.out.println(x: "Invalid!");
68          }
69      }
70  }
71
72  static void addBook() {
73      System.out.print(s: "Title: "); String t = sc.nextLine();
74      System.out.print(s: "Author: "); String a = sc.nextLine();
75      System.out.print(s: "Category: "); String c = sc.nextLine();
76      Book b = new Book(bookId++, t, a, c);
77      books.put(b.bookId, b);
78      saveData();
79      System.out.println("Book added with ID " + b.bookId);
80  }
81
82  static void addMember() {
83      System.out.print(s: "Name: "); String n = sc.nextLine();
84      System.out.print(s: "Email: "); String e = sc.nextLine();
85      Member m = new Member(memberId++, n, e);
86      members.put(m.memberId, m);
87      saveData();
88      System.out.println("Member added with ID " + m.memberId);
89  }
90
91  static void issueBook() {
92      System.out.print(s: "Book ID: "); int bId = sc.nextInt();
93      System.out.print(s: "Member ID: "); int mId = sc.nextInt();
94      if (!books.containsKey(bId) || !members.containsKey(mId)) { System.out.println(x: "Invalid ID
95      Book b = books.get(bId);
96      if (b.isIssued) { System.out.println(x: "Already issued"); return; }
97      b.isIssued = true;
98      members.get(mId).issuedBooks.add(bId);
99      saveData();
100     System.out.println(x: "Book issued successfully");
101 }
102
103 static void returnBook() {
104     System.out.print(s: "Book ID: "); int bId = sc.nextInt();
105     System.out.print(s: "Member ID: "); int mId = sc.nextInt();
106     if (books.containsKey(bId) && members.containsKey(mId)) {
```

```
106             if (books.containsKey(bId) && members.containsKey(mId)) {
107                 books.get(bId).isIssued = false;
108                 members.get(mId).issuedBooks.remove((Integer)bId);
109                 saveData();
110                 System.out.println(x: "Returned successfully");
111             } else System.out.println(x: "Invalid IDs");
112         }
113 
114         static void searchBook() {
115             System.out.print(s: "Enter keyword: ");
116             String k = sc.nextLine().toLowerCase();
117             for (Book b : books.values())
118                 if (b.title.toLowerCase().contains(k) || b.author.toLowerCase().contains(k))
119                     b.display();
120         }
121 
122         static void sortBooks() {
123             List<Book> list = new ArrayList<>(books.values());
124             Collections.sort(list);
125             System.out.println(x: "\nSorted Books:");
126             for (Book b : list) b.display();
127         }
128 
129         static void saveData() {
130             try (PrintWriter bw = new PrintWriter(fileName: "books.txt")) {
131                 for (Book b : books.values())
132                     bw.println(b.bookId + "|" + b.title + "|" + b.author + "|" + b.category + "|
132                     bw.println(b.bookId + "|" + b.title + "|" + b.author + "|" + b.category + "|" + b.isI
133             } catch (Exception ignored) {}
134             try (PrintWriter bw = new PrintWriter(fileName: "members.txt")) {
135                 for (Member m : members.values())
136                     bw.println(m.memberId + "|" + m.name + "|" + m.email + "|" + m.issuedBooks);
137             } catch (Exception ignored) {}
138         }
139 }
140 
```

OUTPUT:

```
Welcome to City Library System

1.Add Book  2.Add Member  3.Issue  4.Return  5.Search  6.Sort  7.Exit
Enter choice: 1
Title: Java Programming Mastery
Author: John Smith
Category: Programming
Book added with ID 101

1.Add Book  2.Add Member  3.Issue  4.Return  5.Search  6.Sort  7.Exit
Enter choice: 2
Name: Sophia Soni
Email: sophia@gmail.com
Member added with ID 201
```

```
1.Add Book  2.Add Member  3.Issue  4.Return  5.Search  6.Sort  7.Exit
Enter choice: 3
Book ID: 101
Member ID: 201
Book issued successfully

1.Add Book  2.Add Member  3.Issue  4.Return  5.Search  6.Sort  7.Exit
Enter choice: 5
Enter keyword: java
101 | Java Programming Mastery | John Smith | Programming | Issued: true

1.Add Book  2.Add Member  3.Issue  4.Return  5.Search  6.Sort  7.Exit
Enter choice: 4
Book ID: 101
Member ID: 201
```

```
Returned successfully

1.Add Book  2.Add Member  3.Issue  4.Return  5.Search  6.Sort  7.Exit
Enter choice: 6

Sorted Books:
101 | Java Programming Mastery | John Smith | Programming | Issued: false

1.Add Book  2.Add Member  3.Issue  4.Return  5.Search  6.Sort  7.Exit
Enter choice: 7
Exiting program...
```