Name- Mritunjaya chauhan

RollNo- 2401420024

Course - B.Tech CSE + Data Science.

Java Assignment-4

#library City Digital Management System.

```
import java.util.*;
import java.~~util.*~~ io.*;
public class LibrarySystem{
    static class Book implements Comparable<Book> {
        int BookId;
        String title;
        String author;
        String category;
        boolean isIssued;
        Book(int bookId, String title, String author, String category,
          boolean isIssued){
            this.bookId = bookId;
            this.title = title;
            this.author = author;
            this.category = category;
            this.isIssued = isIssued;
        }
        void display() {
            System.out.println("ID: " + bookId + " | "
           + title + " | " + author + " | " + category +
           " | Issued: "+ isIssued);
        }

        @Override
        public int compareTo(Book b) {
            return this.title.compareToIgnoreCase(b.title);
        }
    }
  < static class Member {
        int memberId;
        String name;
        String email;
        List<Integer> issuedBooks= new ArrayList<>();

        Member(int memberId, String name, String email){
            this.memberId = memberId;
            this.name = name;
            this.email = email;
        }
        void display(){
            Sout("Member ID: "+ memberId + " | Name: "+
                   name +" | Email: " + email);
        }
    }
```

```
static Map<Integer, Book> books = new HashMaps<> ();
static Map<Integer, Member> members = new Hashmap<>();

static file bookFile = new file( "books.txt");
static File memberFile = mew file ("members.txt");

// Load Data
static void loadFromFile(){
    try {
      if (bookFile.exists()){
        BufferedReader br = new BufferedReader
                    (new FileReader(bookFile));
        String line;
        while ((line= br.readLine()) != null) {
            String d[] = line.split(",");
            int id = Integer.parseInt(d[0]);
            books.put(id, new Book(id, d[1], d[2], d[3],
                Boolean.parseBoolean(d[4])));
        }
        br.close;
      }
```

```
        if (memberFile.exists()){
            BufferedReader br = new BufferedReader
                        (new FileReader (memberFile));
            String line;
            while ((line = br.readLine()) != null){
                String d[] = line.split(",");
                int id = Integer.parseInt(d[0]);
                Member m = new Member(id, d[1], d[2]);

                for(int i=3; i<d.length; i++){
                m.issuedBooks.add(Integer.parseInt(d[i]));
                }
                member.put(id, m);
            }
            br.close()
        }
    } catch (Exception e){}
 }
 static void saveToFile(){
    try {
        BufferedWriter bw = new BufferedWriter(
                new FileWriter(bookFile));
        for(Book b: books.values()){
          bw.write(b.bookId + "," + b.title + "," + b.author
          + "," + b.category + "," + b.isIssued);
          bw.newLine();
        }
        bw.close();
    BufferedWriter bw2 = new BufferedWriter(new FileWriter
                                                (memberFile));
      for (Member m : members.values()){
            bw2.write(m.memberId + "," + m.name + "," +
                  m.email);
            for (int id : m.issuedBooks) {
                bw2.write("," + id);
                bw2.newLine();
            }
        bw2.close();
    } catch (Exception e) {
         Sout ("Exception found");
    }  }
  }
static void addBook(Scanner sc) {
      Sout ("Book ID: ");
      int id = sc.nextInt(); sc.nextLine;
      Sout ("Title:");
      String title = sc.nextLine();
      System.out.println("Author: ");
      String author = sc.nextLine();
      Sout ("category : ");
      String category = sc.nextLine();
```

```
        books.put(id, new Book(id, title, author,
                                category, false));
        saveToFile();
        System.out.println("Book added");
    }
    static void addMember(Scanner sc){
        Sout("Member ID:");
        int id = sc.nextInt(); sc.nextLine;
        Sout("Name:");
        String name = sc.nextLine();
        System.out.println("Email: ");
        String email = sc.nextLine();

        members.put(id, new Member(id,name,email));
        saveTofile();
        Sout("Member added");
    }
    static void issueBook(Scanner sc){
        Sout("Book ID:");
        int bid = sc.nextInt;
        Sout("Member ID:");
        int mid = sc.nextInt();
        if (!books.containsKey(bid) || !members.
                            containsKey(mid)){
            Sout("Invalid ID!");
            return;
        }
    Book b = books.get(bid);
    Member m = members.get(mid);

    if (b.isIssued){
        System.out.println("Already issued!");
        return;
    }

    b.isIssued = true;
    m.issuedBooks.add(bid);
    saveToFile();
    System.out.println("Book issued!");
}

static void returnBook(Scanner sc){
    Sout("Book ID: ");
    int bid = sc.nextInt();
    System.out.println("Member ID: ");
    int mid =sc.nextInt();

    Book b = books.get(bid);
    Member m = members.get(mid);
    b.isIssued =false;
    m.issuedBooks.remove(Integer.valueOf(bid));
    saveToFile();
    Sout("Book returned");
}
```

```
static void searchBooks (Scanner sc) {
sc.nextLine();
    System.out.println ("~~Book~~ keyword : ");
    String key = sc.nextLine().to~~lowercase~~LowerCase();

    for (Book b : books.values()) {
        if (b.title.toLowerCase().contains(key) ||
                b.author.toLowerCase().contains(key) ||
                b.category.toLowerCase().contains (key)) {
            b.display();
        }
    }
}


static void sortBooks {
    List<Books> list = new ArrayList<>(books.values());
    Collections.sort(list);
    for(Books b : list){
        b.display();
    }
}
```

```
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    loadFromFile();
    while (true){
        System.out.println("\n City library digital
                                    system");
        System.out.println("1Add Book");
        System.out.println("2. Add Member");
        System.out.println("3. Issue Book");
        System.out.println("4. Return Book");
        System.out.println("5. Search Books");
        System.out.println("6: Sort Books");
        System.out.println("7. Exit");
        System.out.println("Choice!");
        int ch= sc.nextInt();
      switch (ch) {
          case 1: addBook(sc); break;
          case 2: addMember(sc); break;
          case 3: issueBook(sc); break;
          case 4: returnBook(sc); break;
          case 5: searchBooks(sc); break;
          case 6: sortBooks(sc); break;
          case 7: saveToFile();
                  Sout("Goodbye!");
                  return;
          default:
                  Sout("Invalid!");
}   }   }   }
```