21-06-2024

krishnamoorthy

--KRISHNAMOORTHY

**Java Library Management System**

oops

**Introduction**

This assignment involves creating a simple Java application to manage a library system. The system will allow users to add books, manage members, and process borrowing and returning of books. It will demonstrate the use of classes and objects, inheritance and polymorphism, encapsulation, and abstraction in Java.

------------------------------------------------------------------------------------------------------------------

**Classes and Objects Implementation**

import java.util.ArrayList;

import java.util.List;

class Book {

    private String title;

    private String author;

    private String isbn;

    private boolean available;

    public Book(String title, String author, String isbn) {

        this.title = title;

        this.author = author;

        this.isbn = isbn;

        this.available = true;

    }

    public String getTitle() {

        return title;

    }

    public String getAuthor() {

        return author;

    }

    public String getIsbn() {

        return isbn;

    }

    public boolean isAvailable() {

        return available;

    }

    public void setAvailable(boolean available) {

        this.available = available;

    }

    @Override

    public String toString() {

        return "Book{" +

                "title='" + title + '\'' +

                ", author='" + author + '\'' +

                ", isbn='" + isbn + '\'' +

                ", available=" + available +

                '}';

    }

}

class Member {

    private String name;

    private String memberId;

    private List<Book> borrowedBooks;

    public Member(String name, String memberId) {

        this.name = name;

        this.memberId = memberId;

        this.borrowedBooks = new ArrayList<>();

    }

    public String getName() {

        return name;

    }

    public String getMemberId() {

        return memberId;

    }

    public List<Book> getBorrowedBooks() {

        return borrowedBooks;

    }

    public void borrowBook(Book book) {

        borrowedBooks.add(book);

        book.setAvailable(false);

    }

    public void returnBook(Book book) {

        borrowedBooks.remove(book);

        book.setAvailable(true);

    }

    @Override

    public String toString() {

        return "Member{" +

                "name='" + name + '\'' +

                ", memberId='" + memberId + '\'' +

                ", borrowedBooks=" + borrowedBooks +

                '}';

    }

}

class Library {

    private List<Book> books;

    private List<Member> members;

    public Library() {

        books = new ArrayList<>();

        members = new ArrayList<>();

    }

    public void addBook(Book book) {

        books.add(book);

    }

    public void addMember(Member member) {

        members.add(member);

    }

    public boolean borrowBook(String memberId, String isbn) {

        Member member = findMemberById(memberId);

        Book book = findBookByIsbn(isbn);

        if (member != null && book != null && book.isAvailable()) {

            member.borrowBook(book);

            return true;

        }

        return false;

    }

    public boolean returnBook(String memberId, String isbn) {

        Member member = findMemberById(memberId);

        Book book = findBookByIsbn(isbn);

        if (member != null && book != null && member.getBorrowedBooks().contains(book)) {

            member.returnBook(book);

            return true;

        }

        return false;

    }

    private Member findMemberById(String memberId) {

        for (Member member : members) {

            if (member.getMemberId().equals(memberId)) {

                return member;

            }

        }

        return null;

    }

    private Book findBookByIsbn(String isbn) {

        for (Book book : books) {

            if (book.getIsbn().equals(isbn)) {

                return book;

            }

        }

        return null;

    }

    public void listBooks() {

        for (Book book : books) {

            System.out.println(book);

        }

    }

    public void listMembers() {

        for (Member member : members) {

            System.out.println(member);

        }

    }

}

public class Main {

    public static void main(String[] args) {

        Library library = new Library();

        library.addMember(new Member("KD KRISHNA", "MEM001"));

        library.addMember(new Member("KD LIFE", "MEM002"));

        library.borrowBook("MEM001", "ISBN001");

        library.returnBook("MEM001", "ISBN001");

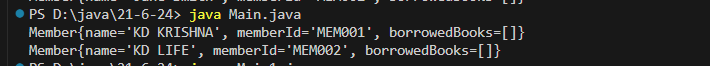
        library.listBooks();

        library.listMembers();

    }

}

------------------------------------------------------------------------------------------------------------------OUTPUT:



------------------------------------------------------------------------------------------------------------------

# Inheritance and Polymorphism

**Premium Member Class**

import java.util.ArrayList;

import java.util.List;

class Book {

    private String title;

    private String author;

    private String isbn;

    private boolean available;

    public Book(String title, String author, String isbn) {

        this.title = title;

        this.author = author;

        this.isbn = isbn;

        this.available = true;

    }

    public String getTitle() {

        return title;

    }

    public String getAuthor() {

        return author;

    }

    public String getIsbn() {

        return isbn;

    }

    public boolean isAvailable() {

        return available;

    }

    public void setAvailable(boolean available) {

        this.available = available;

    }

    @Override

    public String toString() {

        return "Book{" +

                "title='" + title + '\'' +

                ", author='" + author + '\'' +

                ", isbn='" + isbn + '\'' +

                ", available=" + available +

                '}';

    }

}

class Member {

    private String name;

    private String memberId;

    private List<Book> borrowedBooks;

    public Member(String name, String memberId) {

        this.name = name;

        this.memberId = memberId;

        this.borrowedBooks = new ArrayList<>();

    }

    public String getName() {

        return name;

    }

    public String getMemberId() {

        return memberId;

    }

    public List<Book> getBorrowedBooks() {

        return borrowedBooks;

    }

    public boolean borrowBook(Book book) {

        if (borrowedBooks.size() < 5 && book.isAvailable()) {

            borrowedBooks.add(book);

            book.setAvailable(false);

            return true;

        }

        return false;

    }

    public void returnBook(Book book) {

        borrowedBooks.remove(book);

        book.setAvailable(true);

    }

    @Override

    public String toString() {

        return "Member{" +

                "name='" + name + '\'' +

                ", memberId='" + memberId + '\'' +

                ", borrowedBooks=" + borrowedBooks +

                '}';

    }

}

class PremiumMember extends Member {

    public PremiumMember(String name, String memberId) {

        super(name, memberId);

    }

    @Override

    public boolean borrowBook(Book book) {

        if (getBorrowedBooks().size() < 10 && book.isAvailable()) {

            getBorrowedBooks().add(book);

            book.setAvailable(false);

            return true;

        }

        return false;

    }

}

class Library {

    private List<Book> books;

    private List<Member> members;

    public Library() {

        books = new ArrayList<>();

        members = new ArrayList<>();

    }

    public void addBook(Book book) {

        books.add(book);

    }

    public void addMember(Member member) {

        members.add(member);

    }

    public boolean borrowBook(String memberId, String isbn) {

        Member member = findMemberById(memberId);

        Book book = findBookByIsbn(isbn);

        if (member != null && book != null) {

            return member.borrowBook(book);

        }

        return false;

    }

    public boolean returnBook(String memberId, String isbn) {

        Member member = findMemberById(memberId);

        Book book = findBookByIsbn(isbn);

        if (member != null && book != null && member.getBorrowedBooks().contains(book)) {

            member.returnBook(book);

            return true;

        }

        return false;

    }

    private Member findMemberById(String memberId) {

        for (Member member : members) {

            if (member.getMemberId().equals(memberId)) {

                return member;

            }

        }

        return null;

    }

    private Book findBookByIsbn(String isbn) {

        for (Book book : books) {

            if (book.getIsbn().equals(isbn)) {

                return book;

            }

        }

        return null;

    }

    public void listBooks() {

        for (Book book : books) {

            System.out.println(book);

        }

    }

    public void listMembers() {

        for (Member member : members) {

            System.out.println(member);

        }

    }

}

public class Main1 {

    public static void main(String[] args) {

        Library library = new Library();

        library.addBook(new Book("KD KRISHNA", "MOORTHY", "ISBN001"));

        library.addBook(new Book("KD LIFE", "MOORTHY", "ISBN002"));

        library.addMember(new Member("KRISHNA", "MEM001"));

        library.addMember(new PremiumMember("KD KRISHNA", "MEM002"));

        System.out.println("Borrowing Book One by KRISHNA (Regular Member):");

        if (library.borrowBook("MEM001", "ISBN001")) {

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

        } else {

            System.out.println("Failed to borrow the book.");

        }

        System.out.println("Borrowing Book Two by MOORTHY (Premium Member):");

        if (library.borrowBook("MEM002", "ISBN002")) {

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

        } else {

            System.out.println("Failed to borrow the book.");

        }

        System.out.println("\nReturning Book One by KRISHNA:");

        if (library.returnBook("MEM001", "ISBN001")) {

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

        } else {

            System.out.println("Failed to return the book.");

        }

        System.out.println("\nList of all books:");

        library.listBooks();

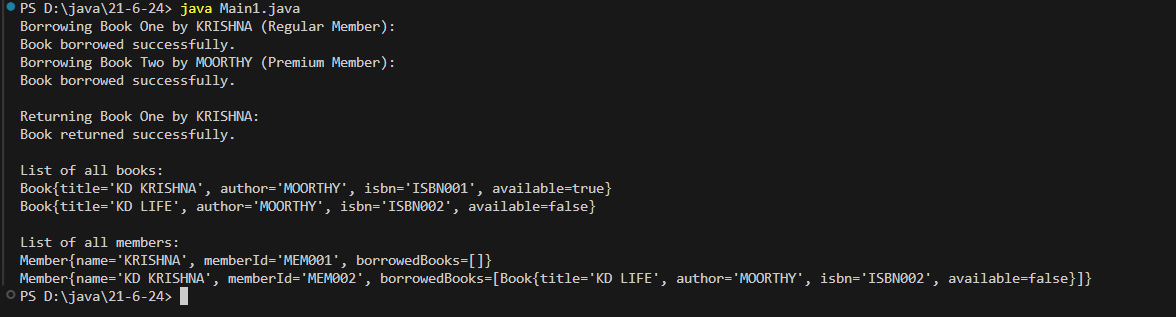
        System.out.println("\nList of all members:");

        library.listMembers();

    }

}

**Output:**

****

**Encapsulation and Abstraction**

* The attributes of each class are encapsulated by making them private and providing public getter and setter methods where appropriate.
* Abstraction is achieved by defining common behaviours between Member and Premium Member through the inheritance hierarchy.

**Code Explanation**

**Design Choices:**

* The Book class encapsulates the details of a book.
* The Member class manages borrowed books and allows for borrowing and returning books.
* The library class manages a collection of books and members and provides methods for adding books and members, as well as borrowing and returning books.
* The Premium Member class extends Member to allow premium members to borrow more books than regular members.

**OOP Principles Applied:**

* **Encapsulation**: Private fields with public getters and setters.
* **Inheritance**: Premium Member inherits from Member.
* **Polymorphism**: Overriding the borrow Book method in Premium Member.
* **Abstraction**: Common behaviours are defined in the Member class and specialized in Premium Member.

**Conclusion**

This assignment demonstrated the application of object-oriented programming principles such as encapsulation, inheritance, polymorphism, and abstraction in a Java program designed to manage a library system. By creating classes for Book, Member, Premium Member, and Library, the program encapsulates the data and behaviours related to books and members, facilitating the management of library operations.