Library Management System

Done By : Anbu Chelvan Francis Vimalan

Description:

The Library Management System (LMS) is a software application designed to manage the operations of a library efficiently. It provides functionalities for librarians to handle tasks such as adding, removing, and updating books in the library's collection.

Features:

- **Add Book:** Librarians can add new books to the library's collection by providing the title and author of the book.
- **Remove Book:** Books can be removed from the library's collection by specifying the book to be removed.
- **View Book Collection:** The system allows users to view the current collection of books in the library.

Benefits:

- **Efficient Book Management:** The LMS allows librarians to manage the library's collection of books efficiently, making it easier to add, remove, and update books as needed.
- **Improved Accessibility:** Users can easily view the available books in the library's collection, facilitating access to information.
- **Simplified Testing:** The `LibraryTester` class provides a convenient way to test the functionalities of the LMS, ensuring that it operates as expected.

LMS Structure:

Classes for the LMS program:

- **Book:** Book’s Title and Author.
- **LMS:** Library management system to add or remove books and also to get storage information.
- **LibraryTester:** Test the LMS by adding books and removing them.

Class Book

This is the `Book` class with the specified attributes `title` and `author`. It also includes constructors, getters, setters, and overrides `equals`, `hashCode`, and `toString` methods for proper comparison and printing.

```java
package midterm.anbu_vimalan_1.task2;
import java.util.Objects;

public class Book {
   private String title;
   private String author;

   // Constructor
   public Book(String title, String author) {
       this.title = title;
       this.author = author;
   }

   // Getters and Setters
   public String getTitle() {
       return title;
   }

   public void setTitle(String title) {
       this.title = title;
   }

   public String getAuthor() {
       return author;
   }

   public void setAuthor(String author) {
       this.author = author;
   }

   // Override equals and hashCode for proper comparison
   @Override
   public boolean equals(Object o) {
       if (this == o) return true;
```

```
        if (o == null || getClass() != o.getClass()) return false;
        Book book = (Book) o;
        return Objects.equals(title, book.title) &&
Objects.equals(author, book.author);
    }

    @Override
    public int hashCode() {
        return Objects.hash(title, author);
    }

    // Override toString for printing
    @Override
    public String toString() {
        return "Book{" +
                "title='" + title + '\'' +
                ", author='" + author + '\'' +
                '}';
    }
}
```

Class LMS

This is the `LMS` class with a `storage` list to hold the books. It includes methods to add a book to the storage, remove a book from the storage, and print the contents of the storage.

```
package midterm.anbu_vimalan_1.task2;
import java.util.ArrayList;
import java.util.List;

public class LMS {
    private List<Book> storage;

    // Constructor
    public LMS() {
        this.storage = new ArrayList<>();
    }

    // Add a book to the storage
    public void addBook(Book book) {
        storage.add(book);
    }
    // Remove a book from the storage
    public boolean removeBook(Book book) {
        return storage.remove(book);
    }

    // Print the contents of the storage
    public void printStorage() {
        for (Book book : storage) {
            System.out.println(book);
        }
    }
}
```

Class LibraryTester

This `LibraryTester` class creates several `Book` objects, adds them to an `LMS` object, prints the contents of the library, removes one of the books, and then prints the updated contents of the library.

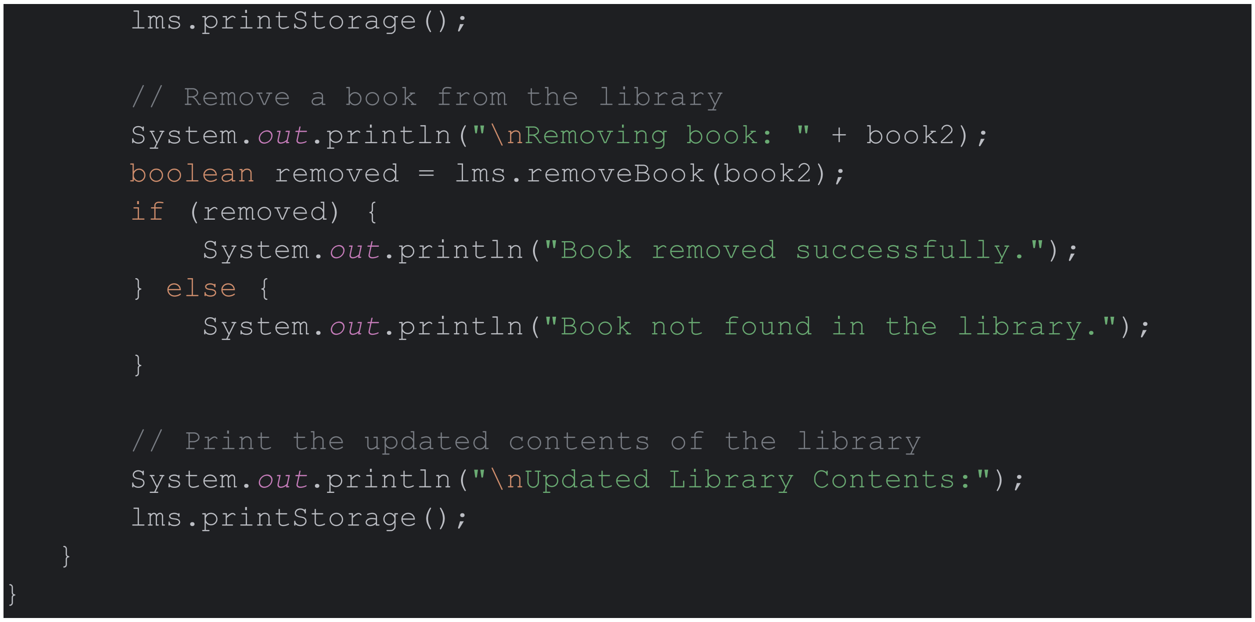
```
package midterm.anbu_vimalan_1.task2;

public class LibraryTester {
    public static void main(String[] args) {
        // Create some Book objects
        Book book1 = new Book("The Post Office", "Rabindranath
Tagore");
        Book book2 = new Book("Malgudi days", "R K Narayan");
        Book book3 = new Book("The private life of an indian prince",
"Mulk Raj Anand");

        // Create an LMS object
        LMS lms = new LMS();

        // Add the books to the library
        lms.addBook(book1);
        lms.addBook(book2);
        lms.addBook(book3);

        // Print the contents of the library
        System.out.println("Library Contents:");
```



```
        lms.printStorage();

        // Remove a book from the library
        System.out.println("\nRemoving book: " + book2);
        boolean removed = lms.removeBook(book2);
        if (removed) {
            System.out.println("Book removed successfully.");
        } else {
            System.out.println("Book not found in the library.");
        }

        // Print the updated contents of the library
        System.out.println("\nUpdated Library Contents:");
        lms.printStorage();
    }
}
```