

**《程序开发实践》项目报告**

**Project Report on Programming in Applications**

**媒体库管理系统**

|  |  |
| --- | --- |
| 学 院 | 东北大学悉尼智能科技学院 |
| 专 业 | 计算机科学与技术 |
| 班级序号 | 20240212 |
| 学 号 | **202410936** |
| 姓 名 | 钱畅阳 |
| 指导教师 | 王和兴 |
| 验收日期 | 2025年5月21日 |

核心知识点清单，由学生确认

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数组 | GUI | 事件处理 | 文件读写 | 默认参数 | 多态 | 多文件/包 | 类 | 泛型 | 虚函数 | Swing | 重载 | 继承/接口 |
| √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |

以下为教师评分表，学生不可填写

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 程序质量  （60%） | 课程设计报告（20%） | 答辩效果  （20%） | 总分 | 等级 |
|  |  |  |  |  |

评分标准与说明：

1. 程序质量（60%）包含程序正确性与所用知识点数量（40%），代码可读性（10%）与界面友好性（10%）。
2. 课程设计报告（20%）要求排版规范，模块设计有文字说明，图、表、代码清单要有序号和名字。
3. 现场答辩（20%）要求根据学生制作的PPT、讲述清晰、回答问题等情况综合评分。
4. 要求学委收集所有学生的代码工程、报告电子版和答辩PPT以备存档。

目录

[1 课题背景 1](#_Toc155692602)

[2 题目分析 4](#_Toc155692603)

[3类结构设计 5](#_Toc155692604)

[4 工程结构组织 8](#_Toc155692605)

[5 核心模块与代码解释 9](#_Toc155692606)

[6 测试与运行 14](#_Toc155692607)

[7 总结 17](#_Toc155692608)

[8 参考文献 18](#_Toc155692609)

# 1 课题背景

**1、问题描述：**

为了方便管理图书馆的资料流通，可以设计一个媒体库管理系统。图书馆的资料主要分为三类：图书、视频光盘和图画。

共同属性: 这三类资料都有以下基本信息：

编号：唯一标识

标题：资料名称

作者：创作者

评级：分为未评级、一般、成人、儿童

各类专属信息:

图书：出版社：出版单位，ISBN号：国际标准书号，页数：书的总页数。

视频光盘：出品者：制作单位，出品年份：制作年份 (使用自定义 Time 类处理日期)，视频时长：播放时长（分钟）。

图画：出品国籍：创作者所属国家，尺寸：长和宽（厘米）。

这个系统让图书馆的管理更轻松，借阅和查找也会更高效。与此同时，为了更好地管理用户，设计一个用户管理系统也是很有必要的。本系统采用 Java 语言和 Swing 图形库进行开发，旨在提供一个用户友好的图形界面。

**2.功能要求：**

为了满足图书馆实际需求和用户需要，将系统分为以下五个模块：

模块一：用户管理模块

功能：管理用户的登录、注册、注销、删除等功能。确保用户在登录后才能使用媒体库管理系统。

代码实现：

user 包：包含 User (抽象基类)、Administrator、RegularUser 类，用于定义用户属性和角色。

UserManager 类：管理用户信息（用户名、密码、角色），提供登录、注册、注销、删除等功能，并检查用户的登录状态。

意义：确保用户身份验证和权限控制，使得系统能够区分普通用户和管理员。控制用户登录状态，从而限制对媒体库管理功能的访问权限。

模块二：媒体库物品管理模块

功能：管理图书、视频光盘、图画等不同类型的物品。支持对物品的增删改查操作。

代码实现：

media 包：包含 Object (抽象基类) 和子类 Book、VideoDisk、Picture。Object 类提供物品的基本信息及通用操作接口。子类分别继承自基类并扩展其特有的属性和功能。Time 类用于处理日期。

LibraryManager 类：

addItem(media.Object item)：添加物品，支持分类并检查编号重复。

deleteItemById(String id)：通过物品编号删除指定物品。

editItemById(String id, media.Object updatedItemData)：修改指定编号的物品

getItems(int typeFilter)：获取当前物品库内容，支持按类别和查询次数排序展示。

queryItem(String query, boolean byId)：根据物品编号或标题查询物品，并记录查询次数。

意义：实现了媒体库中物品的核心管理功能。通过继承和多态实现不同物品类型的统一管理，增强了系统的可扩展性和灵活性。

模块三：推荐算法模块

功能：根据物品的查询次数进行排序，实现简单的推荐算法。

代码实现：

LibraryManager.sortByQueryCount()：内部使用 java.util.Comparator 和 List.sort() 方法，通过物品的 queryCount 属性对物品列表进行降序排序，将查询次数多的物品排在前面。

意义：通过查询次数的排序来实现推荐功能，帮助用户快速找到最受欢迎的物品。

模块四：文件存取模块

功能：将物品数据保存到文件，也支持从文件中加载物品数据，实现数据的持久化。

代码实现：

media.Object 及其子类的 saveToFile(PrintWriter writer) 和 loadFromFile(Scanner scanner) 方法：每个物品对象负责自身的序列化和反序列化逻辑，使用 | 作为字段分隔符。

LibraryManager 中的 saveToFile() 和 loadFromFile() 方法：使用 java.io.PrintWriter 和 java.util.Scanner 配合文件流，遍历物品库调用每个物品的存取方法，并处理文件读写异常。

意义：实现数据的持久化存储和加载，使得媒体库在程序退出后能够保留物品数据。

模块五：主程序与**GUI**控制模块

功能：控制整个程序的流程，包括用户管理和媒体库管理两个系统的切换。提供用户友好的图形界面 (GUI)，接受用户输入并通过界面元素执行相应操作。

代码实现：

Main.java：应用程序入口，初始化 UserManager、LibraryManager 和主 GUI 窗口 LibraryAppGUI。

gui 包：包含所有 Swing 组件类，如 LibraryAppGUI (主框架)、LoginPanel、MainMenuPanel、LibraryDisplayPanel、UserPanel 以及各种操作的对话框 (AddItemDialog, EditItemDialog, RegisterUserDialog)。

LibraryAppGUI 使用 CardLayout 管理不同视图面板的切换。

GUI 组件通过事件监听器响应用户操作，并将业务逻辑委托给 UserManager 和 LibraryManager 处理。

意义：作为整个程序的控制中心和用户交互界面，负责用户交互和各个功能模块的调度，增强程序的可操作性和用户体验。

问题的解决方案：

**3.根据系统功能要求，可以将问题解决分为以下步骤：**

1.应用系统分析，建立该系统的功能模块框图以及 GUI 界面的组织和设计；

2.分析系统中的各个实体及它们之间的关系；

3.根据问题描述，设计系统的类层次结构；

4.完成类层次中各个类的描述（属性和方法）；

5.完成类中各个成员函数的定义；

6.完成系统的应用模块（核心业务逻辑）；

7.设计并实现图形用户界面 (GUI) 模块；

8.功能调试与集成测试；

9.完成系统总结报告。

# 2 题目分析

根据题目要求，本课设主要的要求分析如下：

问题分析

为了提高图书馆资料的管理效率，设计一个媒体库管理系统是必要的。该系统需要管理图书、视频光盘、图画三大类资料，每类资料都有共同属性和特有属性：

共同属性：编号、标题、作者、评级（未评级、一般、成人、儿童）。

图书：出版社、ISBN号、页数。

视频光盘：出品者、出品年份、视频时长。

图画：出品国籍、尺寸（长宽）。

此外，用户管理系统是不可或缺的，能够提供用户注册、登录、注销、权限管理等功能，确保只有合法用户能够访问媒体库。

功能需求：

资料管理：支持添加、删除、修改、查询三类资料，并根据查询次数推荐热门资料。

用户管理：提供用户权限控制，区分普通用户和管理员的操作权限。

数据存储：支持资料和用户信息的持久化存储。

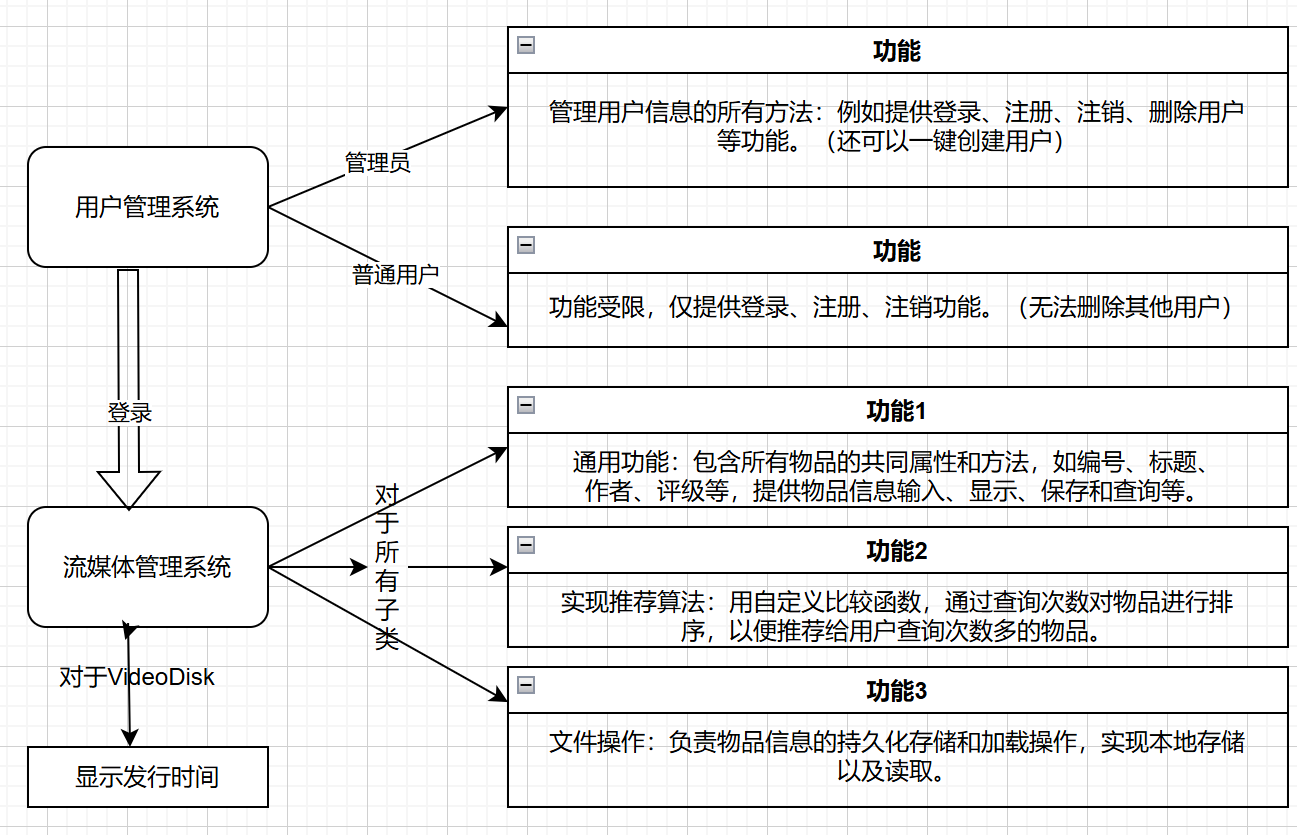
系统目标：

高效管理：简化资料流通，提高查找和借阅效率。

提升用户体验：智能推荐和分类展示资料，提升用户查找效率。

系统可维护性：持久化数据存储，易于扩展和维护。

通过这些设计，可以确保图书馆管理的高效性、安全性和可扩展性。



**GUI界面**

系统功能结构图

# 3类结构设计

1.用户相关类 (user 包)：

User (抽象类)：系统的基础用户类，包含用户名、密码、角色等基本信息。

Administrator 和 RegularUser 类：均继承自 User 类，分别表示管理员和普通用户。

UserManager 类：负责管理所有用户信息，如创建、删除用户、登录验证等。它与 User 类是组合关系。

2.物品相关类 (media 包)：

Object (抽象类)：物品的基类，包含所有物品的共同属性（编号、标题、作者、评级、查询次数）和抽象方法（如 saveToFile, loadFromFile, getDisplayString）。

Book, Picture, VideoDisk 类：均继承自 Object 类，分别代表图书、图画、视频光盘，并包含各自的特有属性和方法。

Time 类：用于表示日期，被 VideoDisk 类组合使用。

3.媒体库管理类 (library 包)：

LibraryManager 类：负责管理 media.Object 对象的集合（如图书、图片、视频光盘）。它与 media.Object 类是聚合关系。它也包含了文件读写、排序推荐等核心逻辑。

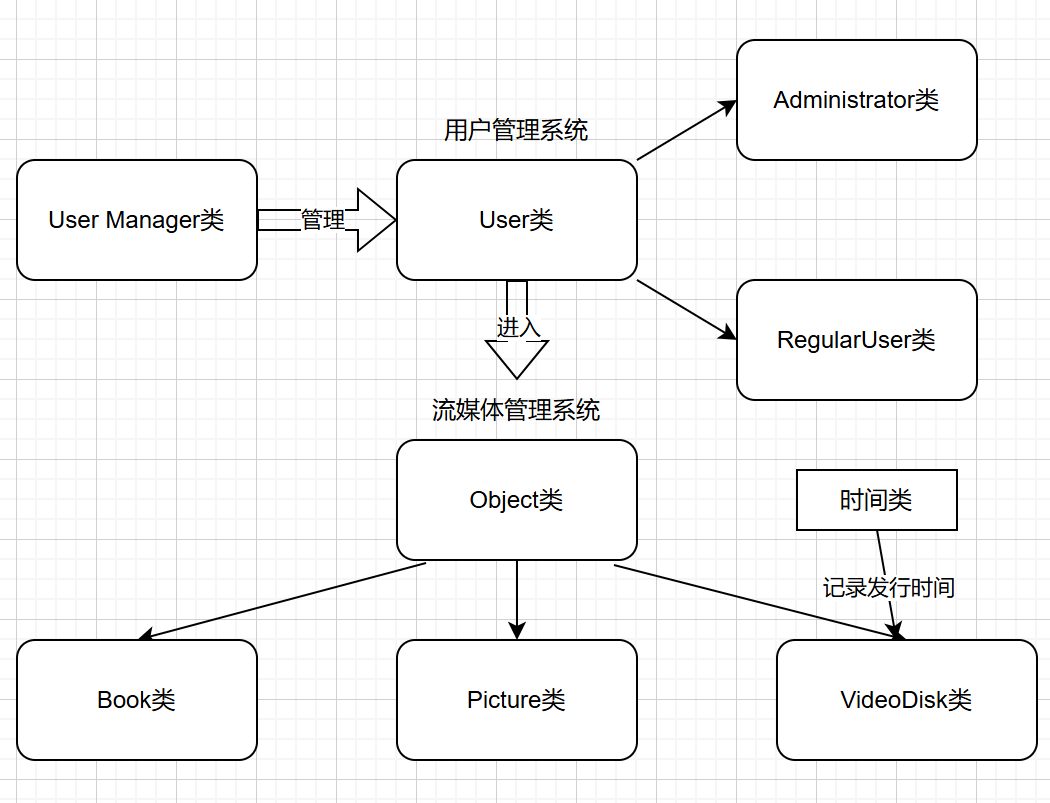
4.GUI 相关类 (gui 包)：

LibraryAppGUI (继承 JFrame)：应用程序的主窗口框架，使用 CardLayout 管理其他面板。

LoginPanel, MainMenuPanel, LibraryDisplayPanel, UserPanel (均继承 JPanel)：构成主窗口的不同视图。

AddItemDialog, EditItemDialog, RegisterUserDialog (均继承 JDialog)：用于执行特定操作的模态对话框。

这些 GUI 类与 UserManager 和 LibraryManager 交互，将用户操作转化为业务逻辑调用，并展示结果。各类之间的继承关系如图所示



**gui包**

**（Swing）**

类之间的继承关系

类功能简述：

User 及其子类 (Administrator, RegularUser)：定义用户模型，包含用户名、密码、角色和密码验证。

UserManager: 封装用户注册、登录、注销、删除等逻辑，并管理当前登录用户状态。

Object 及其子类 (Book, VideoDisk, Picture): 定义物品模型，包含通用属性和各自特有属性，以及文件存取和显示方法。

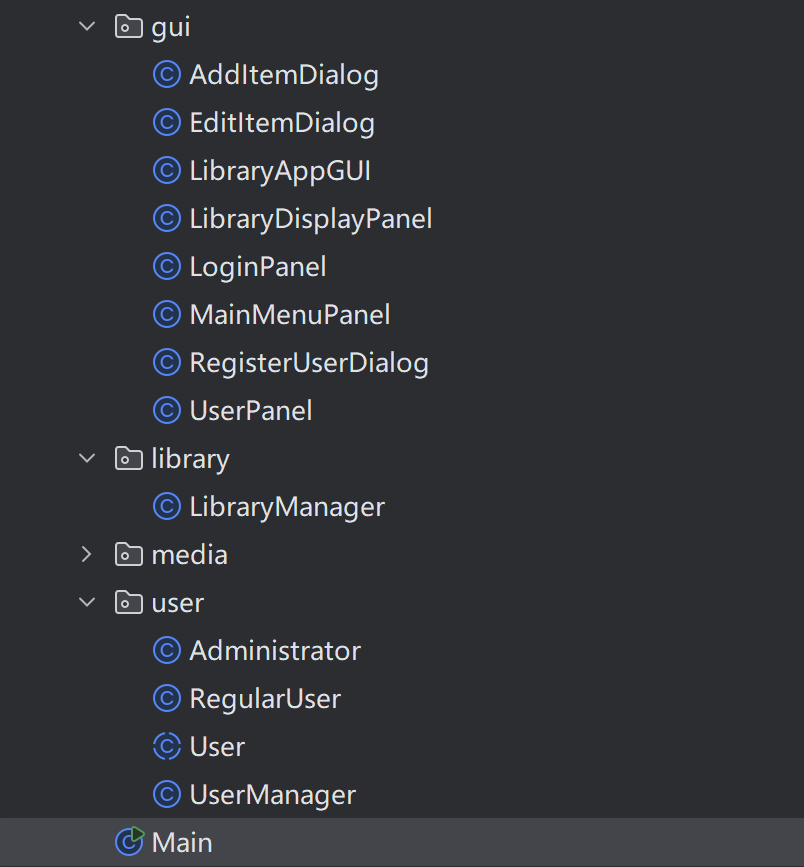
Time: 封装日期处理逻辑，提供日期解析、格式化功能，增强日期输入的健壮性。

LibraryManager: 封装媒体库物品的增删改查、文件持久化、排序推荐等核心业务逻辑。

GUI 类 (如 LibraryAppGUI, LoginPanel, LibraryDisplayPanel, 对话框等): 负责构建用户界面，捕获用户输入事件，调用 Manager 类的方法执行操作，并更新界面显示结果。

# 4 工程结构组织

本项目使用IDEA实现，工程文件结构图如图所示



流媒体管理结构图

GUI 设计理念：

主从结构与卡片布局：LibraryAppGUI作为主框架，采用CardLayout管理多个核心面板（LoginPanel, MainMenuPanel, LibraryDisplayPanel, UserPanel）。这种设计使得界面可以在不同状态间平滑切换，用户体验更佳。

职责分离：每个面板和对话框都有明确的职责。例如，LoginPanel只负责登录，LibraryDisplayPanel只负责展示和筛选物品，AddItemDialog只负责添加物品。具体业务逻辑通过调用UserManager和LibraryManager的方法实现。

模态对话框：对于需要用户集中完成的操作（如添加物品、编辑物品、注册用户），使用模态的JDialog，确保用户在完成当前任务前不能操作主界面其他部分。

用户反馈：通过JOptionPane及时向用户显示操作结果（成功、失败、警告信息），引导用户操作。

数据展示：LibraryDisplayPanel使用JTable和DefaultTableModel来展示物品列表，并利用TableRowSorter实现点击表头排序功能。

组件复用与封装：虽然本例中组件复用程度不高，但每个GUI类封装了自己的布局和事件处理，易于理解和维护。

**流媒体管理项目文件功能列表**

|  |  |  |
| --- | --- | --- |
| 序号 | 文件名 | 说明 |
| 1 | LibraryAppGUI.java | 应用程序主窗口，使用CardLayout管理其他面板，协调各部分交互。 |
| 2 | LoginPanel.java | 用户登录界面，包含用户名、密码输入和登录、用户管理按钮。 |
| 3 | MainMenuPanel.java | 登录成功后的主菜单界面，提供各项功能按钮。 |
| 4 | LibraryDisplayPanel.java | 显示媒体库物品列表的面板，使用JTable，支持过滤和排序。 |
| 5 | UserPanel.java | 用户管理界面，显示用户列表，提供注册新用户和删除用户（管理员权限）功能。 |
| 6 | AddItemDialog.java | 添加新物品的模态对话框，根据物品类型动态显示输入字段。 |
| 7 | EditItemDialog.java | 编辑现有物品的模态对话框，预填充物品信息，物品编号不可编辑。 |
| 8 | RegisterUserDialog.java | 注册新用户的模态对话框，包含用户名、密码和角色选择。 |

# 

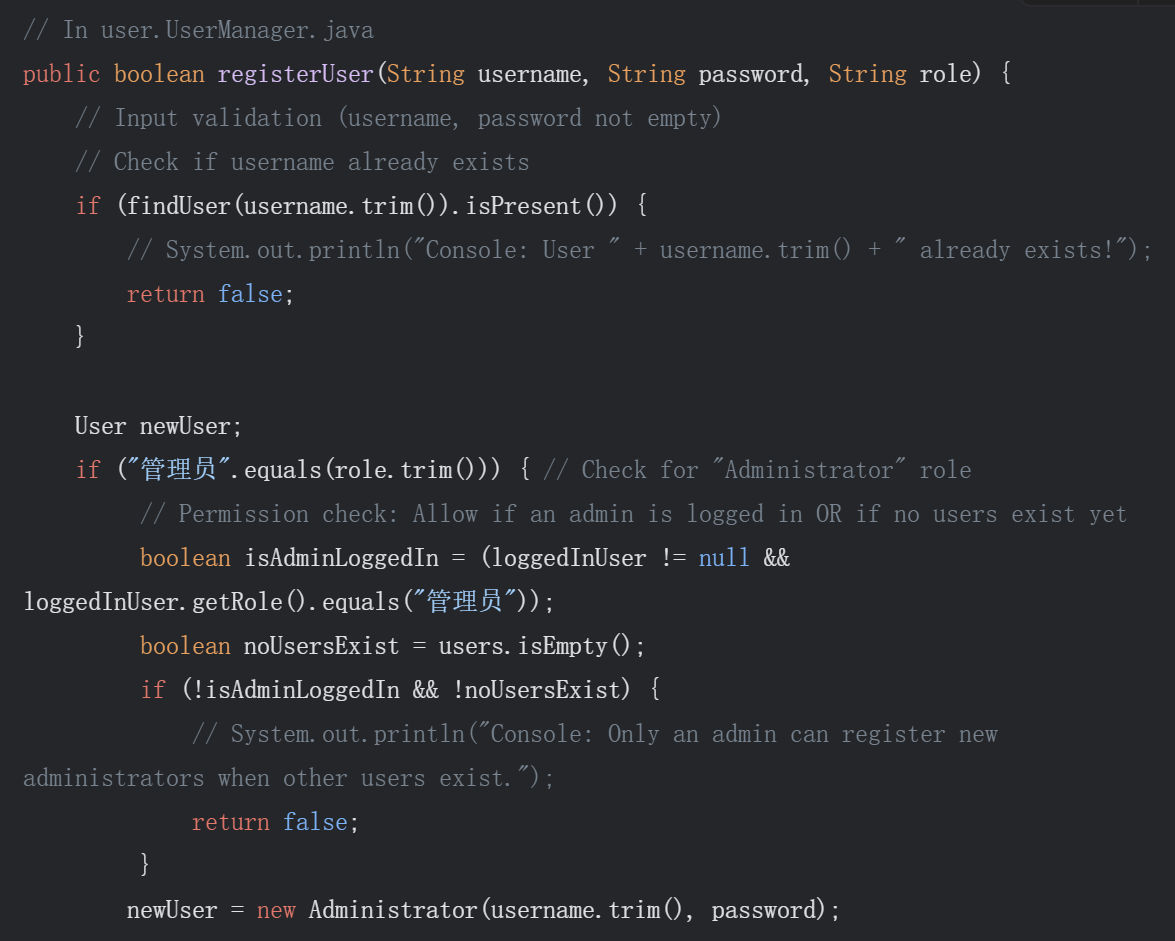
# 5 核心模块与代码解释

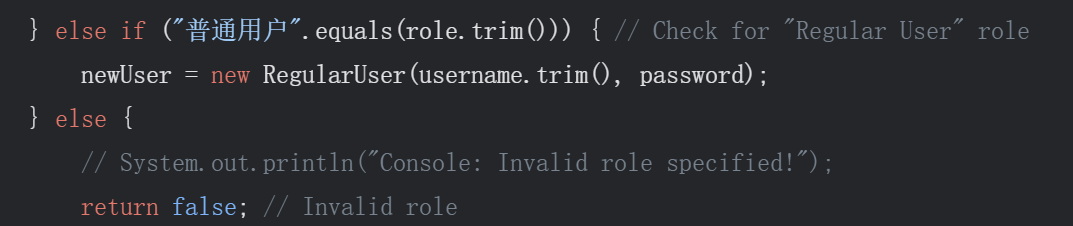
本系统由多个核心模块组成，包括用户管理、物品管理、文件存取和 GUI 交互。以下将展示这些模块的核心代码实现及其与 GUI 的交互方式。

**用户管理模块 (UserManager 与 GUI 交互)**

UserManager 类封装了用户账户相关的业务逻辑。GUI 组件通过调用其方法执行用户操作。

用户注册 (RegisterUserDialog -> UserManager.registerUser)

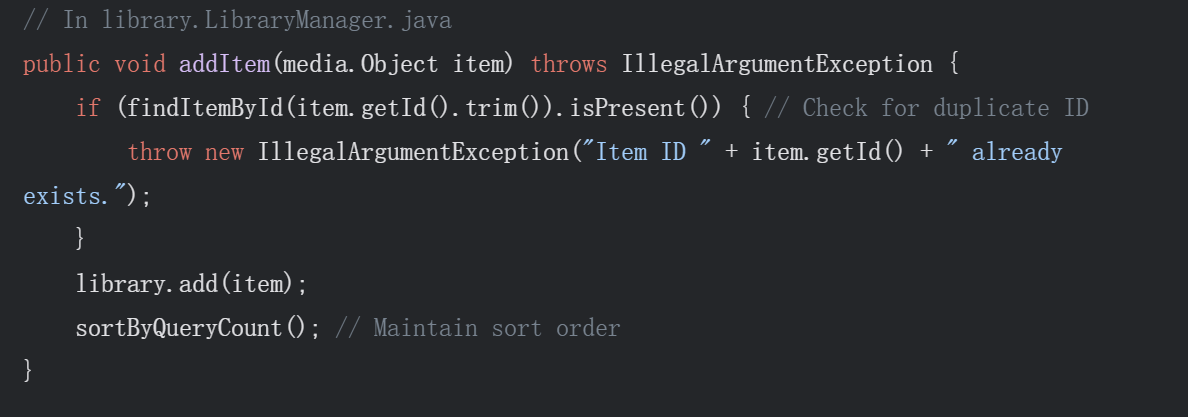




**物品管理模块 (LibraryManager 与 GUI 交互)**

LibraryManager 负责管理媒体库中的物品。

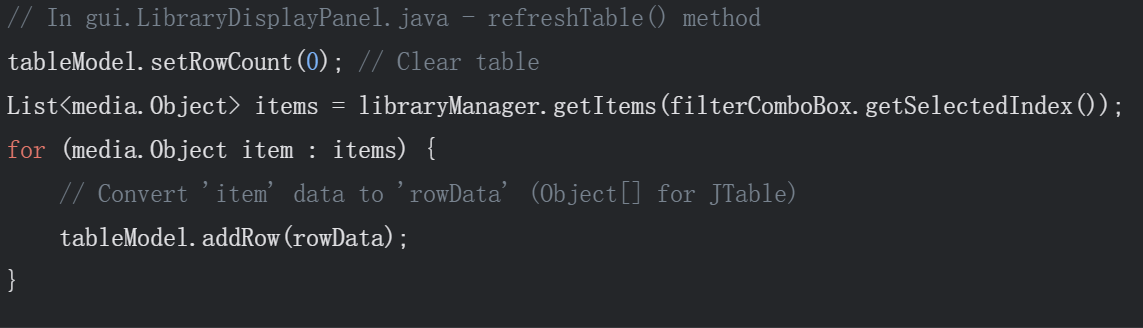
核心逻辑 (LibraryManager.addItem):



**sort自定义排序函数**

显示物品 (LibraryDisplayPanel.refreshTable -> LibraryManager.getItems)

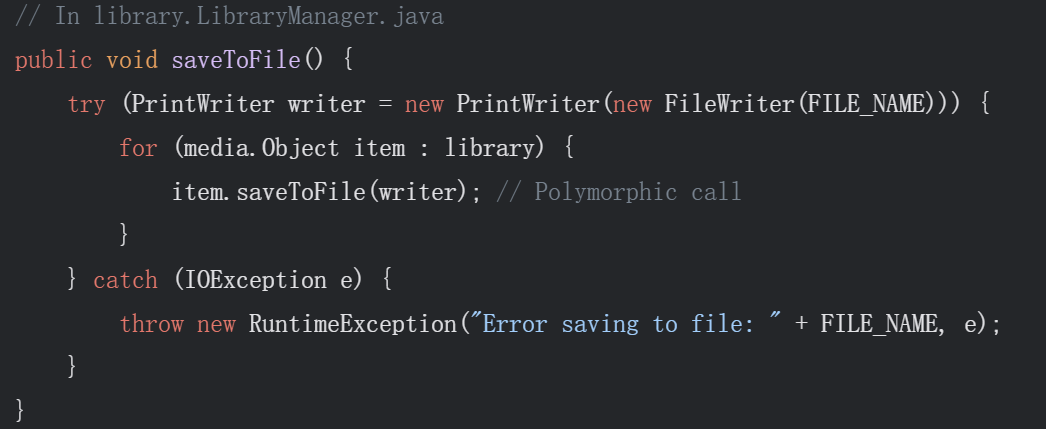
GUI 交互 (关键代码)：



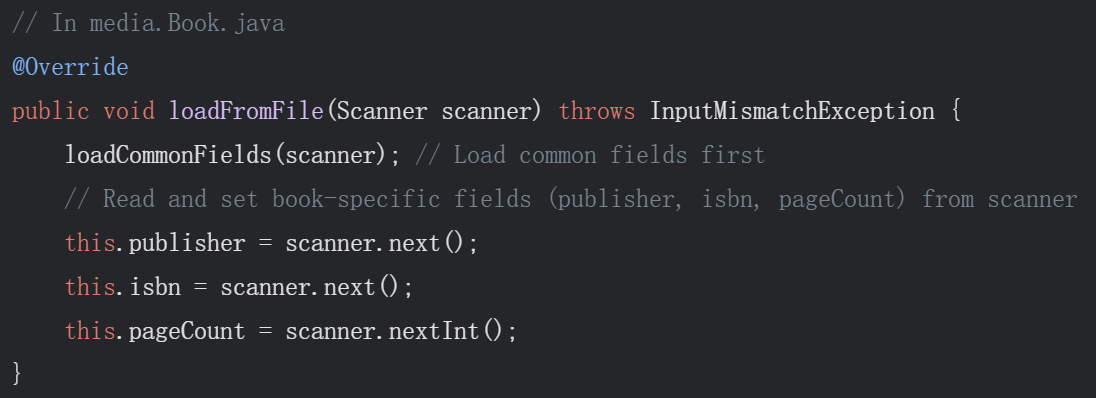
**文件存取模块 (LibraryManager 与 media.Object 子类)**

数据持久化通过 LibraryManager 协调，每个 media.Object 子类负责序列化自身数据。

保存到文件 (LibraryManager.saveToFile 到各 media.Object 子类的 saveToFile)



核心逻辑 (例如 media.Book.loadFromFile):



# GUI (Swing)核心组件与布局

JFrame (LibraryAppGUI): 作为顶级窗口容器。

JPanel: 用于组织和容纳其他组件，构成各个视图（LoginPanel, MainMenuPanel 等）。

CardLayout: 用于在 LibraryAppGUI 的主内容区切换不同的 JPanel，实现单页面应用的效果。

Layout Managers (BorderLayout, FlowLayout, GridBagLayout): 用于在 JPanel 和对话框中自动排列组件，

使界面具有响应性和美观性。

GridBagLayout 被广泛用于对话框中精确控制标签和输入框的对齐。

JTable 和 DefaultTableModel (LibraryDisplayPanel): 用于以表格形式展示媒体库物品，易于阅读和管理。

JList 和 DefaultListModel (UserPanel): 用于展示用户列表。

JButton, JTextField, JPasswordField, JComboBox, JLabel: Swing 的标准组件，用于构建交互元素。

JDialog (AddItemDialog, EditItemDialog, RegisterUserDialog): 用于模态交互，强制用户完成当前操作。

JOptionPane: 用于显示简单的信息、警告、错误对话框和输入对话框。



事件监听器 (Lambda 表达式或匿名内部类):

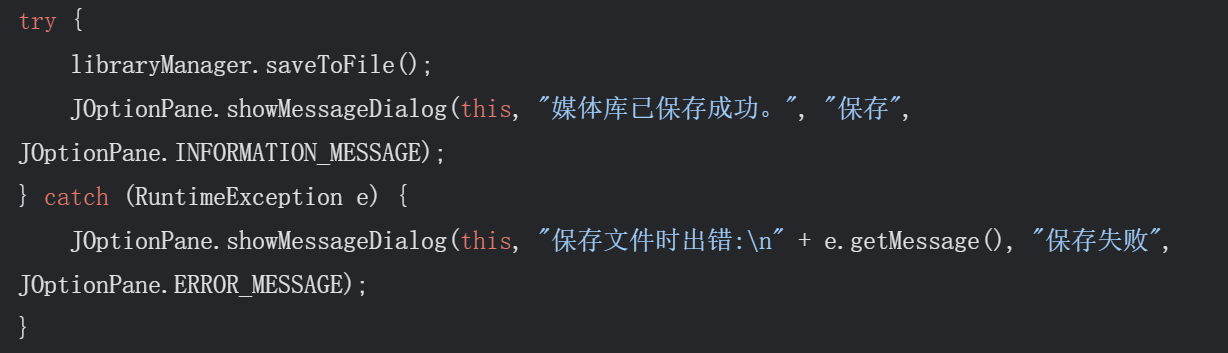


Swing 组件的使用 (JTable, JDialog 等)

Comparator 用于排序 (LibraryManager.sortByQueryCount)



CardLayout 切换面板:



异常处理 (try-catch):

# 6 测试与运行

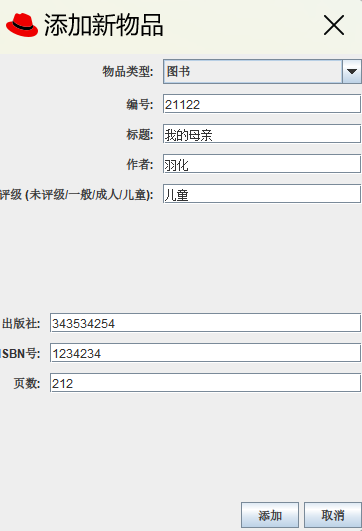
本系统所有模块均进行了单元测试，最后进行进行集成测试，系统运行主要界面如下图所示：



用户管理系统



在登陆后即可进入媒体管理界面



添加物品功能



文档存入功能

**值得注意的是，本程序十分人性化，几乎每一部操作无论正确与错误，都会给予用户充分的提示语引导（实例如上图所示），使得用户就算操作失误也知道自己错在哪里。并且还有一键创建用户的功能和*推荐算法（将查询次数高的排在最前面，告诉用户哪本书最热门）*，使得本程序的使用变得十分方便。**



推荐算法（将查询次数高的排在最前面）

# 7 总结

7 总结

从选题到本课题最终完成，已经过去了22天。这是我第一次使用 Java Swing 完成一个较为完整的桌面应用程序项目。这里首先总结根据课程要求，列出本项目所用到的核心 Java 知识点清单。

表2. 媒体库管理项目 (Java Swing 版) 所涉及知识点总结

项目过程与收获：

通过本次课程设计，我深入学习并实践了 Java 面向对象编程的核心概念，特别是继承、多态和封装在实际项目中的应用。

将原 C++ 项目改写为 Java Swing 版本，让我对两种语言在特性和设计思路上有了更直观的对比。例如，Java 的自动内存管理（垃圾回收）简化了资源管理，但需要关注对象引用的生命周期；Java 没有指针，但通过引用实现类似功能；Java 的单继承机制促使我更多地思考组合和接口设计（尽管本例中主要使用继承）。

GUI 开发体验：

使用 Swing 构建图形用户界面是一个全新的挑战。学习和应用各种 Swing 组件、布局管理器以及事件处理模型，让我对 GUI 程序设计有了初步的认识。

布局管理：GridBagLayout 虽然灵活但配置较为复杂，CardLayout 在实现多视图切换方面非常有效。

事件驱动编程：深刻理解了事件监听和响应机制对于 GUI 程序的重要性。

模型-视图分离：虽然没有严格实现 MVC/MVP/MVVM 等设计模式，但在设计中尽量将界面展示 (gui 包) 与业务逻辑 (UserManager, LibraryManager) 分离，提高了代码的可维护性。

用户体验：认识到提供清晰的用户反馈（如通过 JOptionPane）和直观的操作流程对于提升用户体验至关重要。

主要学习点回顾：

Java 集合框架与 Stream API (如果使用)：ArrayList 用于存储动态数据集合。如果使用了 Stream API（例如在排序或过滤中），则可以体验其声明式编程的便利。

异常处理 (try-catch)：在文件操作和数据转换中，通过 try-catch 块处理潜在的 IOException、NumberFormatException 等，保证了程序的健壮性。

文件 I/O (java.io 和 java.util.Scanner): 掌握了使用 Java I/O 流进行文本文件的读写，实现了数据的持久化。

Swing GUI 组件与事件处理: 熟悉了常用 Swing 组件的用法，以及如何通过事件监听器响应用户交互。

该项目仍然有许多可以改进的地方，例如：

更完善的输入校验和错误提示。

更高级的推荐算法。

可能的用户数据持久化。

使用更现代的 GUI 设计模式或框架

更细致的单元测试。

从这个项目中，我深刻感受到编写代码和调试代码需要耐心和坚持不懈的精神。每当遇到编译错误或运行时问题，通过不断分析、查阅资料和反复调试，最终解决问题的过程充满了挑战，也带来了巨大的成就感。编程不仅仅是技术的堆砌，更是逻辑思维、问题解决能力和细致耐心的综合体现。

# 8 参考文献

8 参考文献

[1] Oracle Java SE Documentation

[2] Stack Overflow https://stackoverflow.com

[3] CSDN网站 https://www.csdn.net