

浙江大学

数据库系统 春学期程序设计实验报告



课程名称：____数据库系统____ 实验类型：____设计开发型____

实验项目：____基于 Java Swing 和 MySQL 的图书管理系统____

学生姓名：____常淮麟____

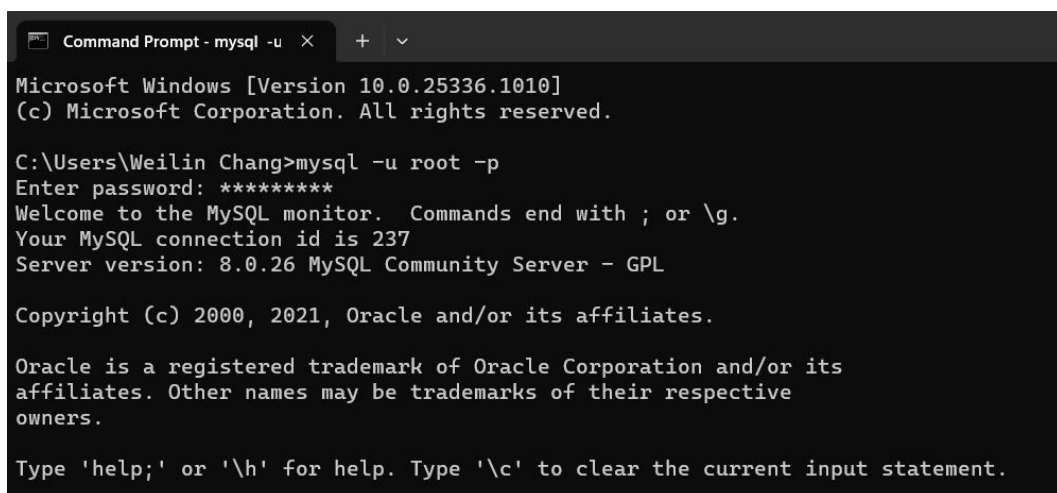
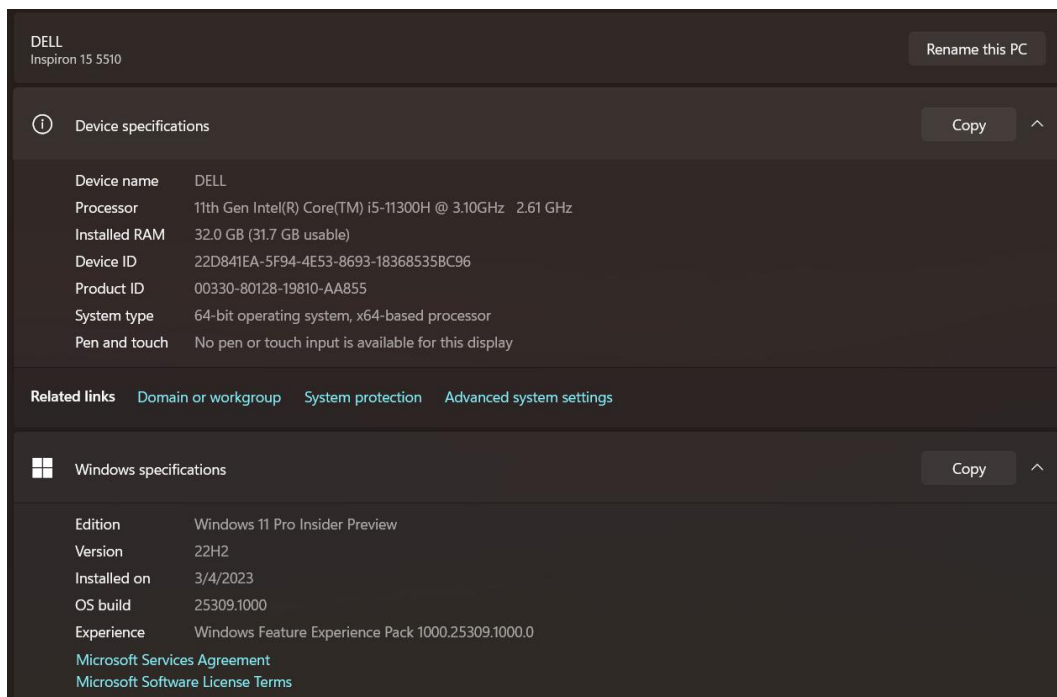
同组学生：____（无）____

指导老师：____陈岭____ 实验地点：____紫金港机房____

验收日期：____2023____年____5____月____14____日

一、 主要仪器设备

实验中使用 x86-64 架构, DELL Inspiron 15 5510 笔记本, 配备第 11 代 Intel Core i5-11300H 四核八线程处理器, 标准运行频率 3.1GHz, 32GB MEM, 512 GB + 1 TB SSD。在 Windows 11 PRO (22H2) Dev 25309.10 版本系统下, 基于 MySQL (8.0.26 版本) 数据库, 使用 Java JDK 17 开发工具包进行开发实验, 使用的集成开发环境为 IntelliJ IDEA Community Edition 2023.3.3。



二、 实验目的和要求

1 实验目的

设计并实现一个精简的图书管理系统，要求具有图书入库、查询、借书、还书、借书证管理等功能。

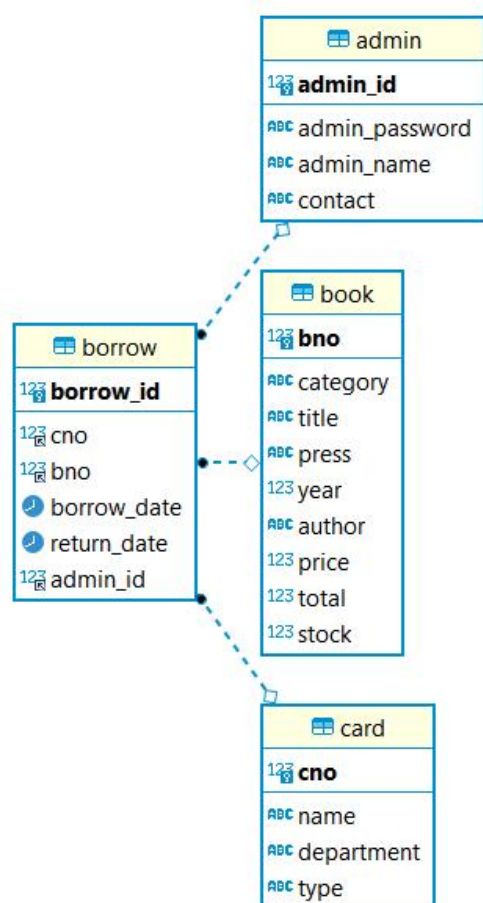
通过该图书馆系统的设计与实现，提高学生的系统编程能力，加深对数据库系统原理及应用的理解。

2 系统需求

(1) 基本数据对象

根据实验需求，我设计如下的 E-R 图（实体-联系图），其中数据库中存储四张表，分别为：book 表（图书表），card 表（借书卡表），borrow 表（借书记录表），admin 表（管理员表）。

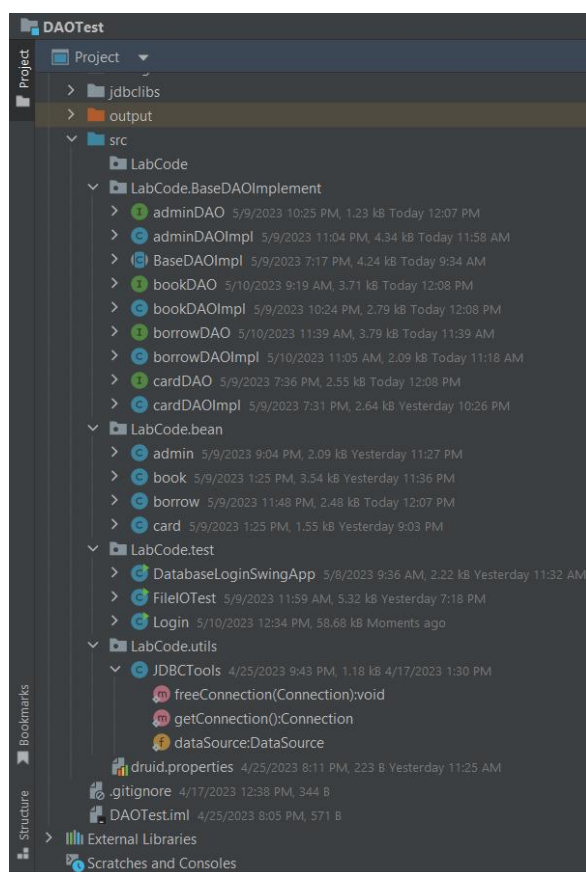
并根据具体情况为每张表设置一个连续自增长的业务主键。



对象名称	包含属性
书	书号（主键），类别，书名，出版社，年份，作者，价格，总藏书量，库存
借书证	卡号（主键），姓名，单位，类别（教授，职工，本科生，研究生）
管理员	管理员 ID（主键），密码，姓名，联系方式
借书记录	记录号（主键），卡号，借书证号，借出日期，归还日期，经手人（管理员 ID）

（2）基本功能模块：

根据实验需求，我在项目中设置了如下的代码组织形式。



其中按包进行代码组织，可以分为三部分：bean 包，util 包，dao 包，test 包

（1）bean 包用于封装四张表和表的一行记录的类

（2）util 包中包含 JDBCTools 类，封装了数据库连接的获取和释放的方法

（3）dao 包是整个项目代码的核心，目的是将数据库的一条记录转化为 Java 中的对象，同时实现了对数据库表的增删改查方法。其中包含了通用的 DAO 实现类

BaseDAOImpl, 并为每一个 Javabean 接口都编写了一个 DAO 类及其对应的接口实现类。（如：bookDAOImpl 类负责 book 表与 book 基于 sql 语句的映射关系）

（4）test 包主要包含了与用户交互的界面与登录程序,同时对 dao 包进行测试。

```

bean包
    Department
    Employee
    用于封装t_department和t_employee表的一行记录的类
util包
    JDBCTools
    封装了数据库连接的获取和释放的方法
    getConnection() 获取一个数据库连接
    freeConnection() 释放一个连接

dao包
    把数据库的一条记录 <--> Java对象
    增、删、改    把Java对象-->一条记录
    查            记录-->Java对象
    查询多个对象
    查询单个对象
    查询其他值

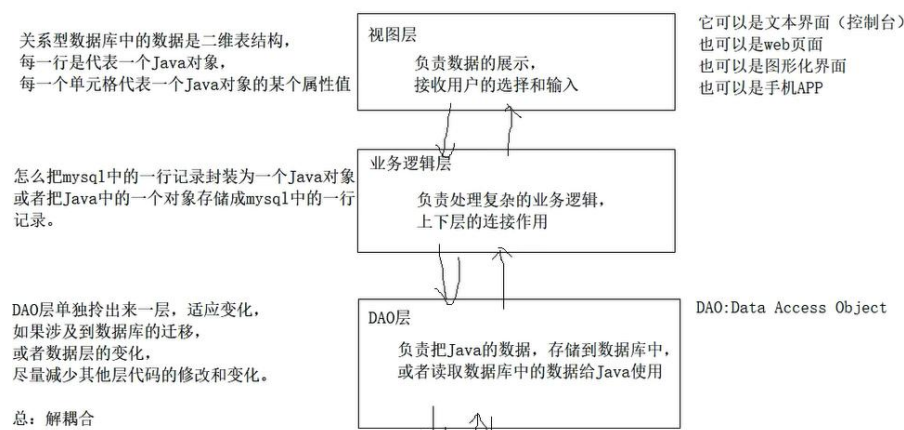
通用DAO的实现类BaseDAOImpl
给每一个Javabean对应了一个DAO类
DepartmentDAO接口和DepartmentDAOImpl类
    
```

对于数据库管理系统,为了实现功能分块和解耦合,我将其按照功能实现的不同分为三层:视图层、业务逻辑层、DAO 层 (Data Access Object)。

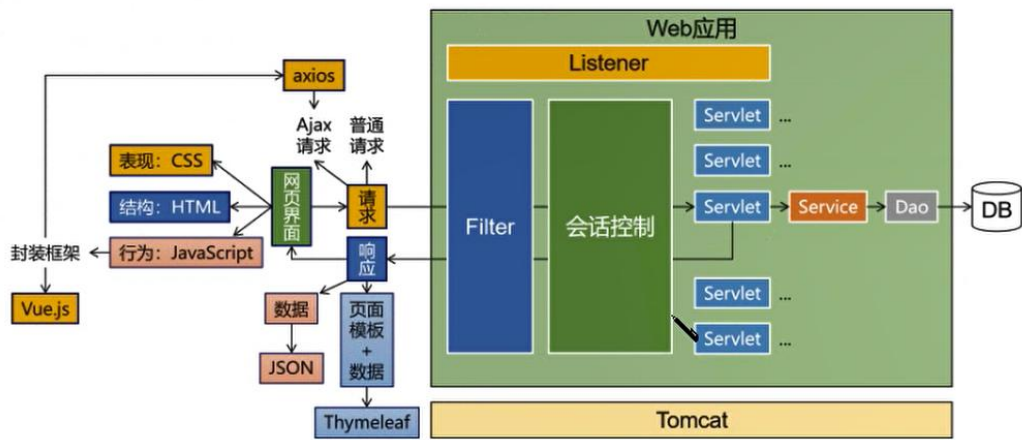
视图层负责数据的展示,并接受用户的操作,在本实验中我使用控制台命令的形式完成对数据库的操作,实际上也可使用图形化界面(浏览器或客户端)。

业务逻辑层负责表中的一行记录与 Java 对象的相互转换,起到连接 DAO 层和视图层的作用。

DAO 层负责把 Java 中与用户交互后的数据以特定的格式存储到数据库中,或者是根据用户的需求从数据库在读取数据。因此叫做“数据访问对象”层。

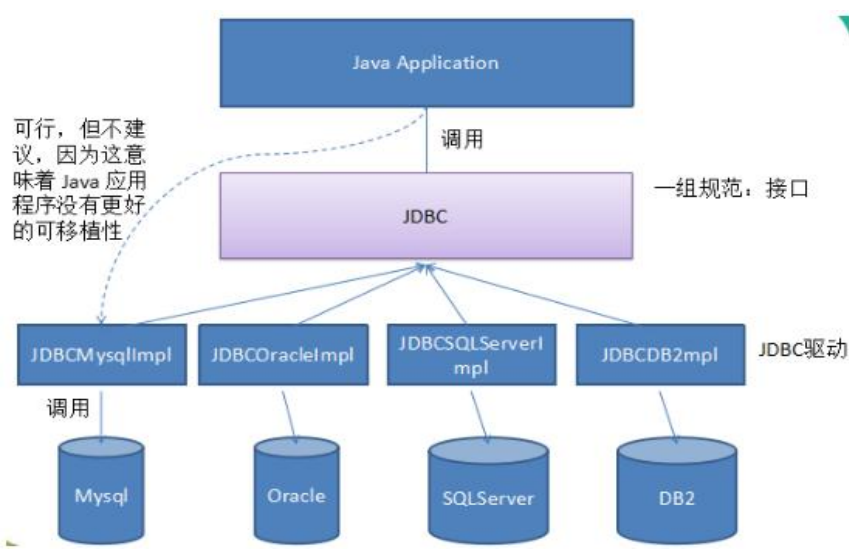


下面是基于 B/S 框架的数据库程序体系结构：其中可使用 Vue 架构实现浏览器前端的总体布局，通过 Tomcat 部署 Web 应用程序，经由 DAO 层访问数据库。



(3) 专用术语解释：

1. JDBC：Java Database Connectivity，它是代表一组独立于任何数据库管理系统（DBMS）的 API，声明在 java.sql 与 javax.sql 包中，是 Oracle 公司提供的一组接口规范，由各个数据库厂商来提供实现类，这些实现类的集合构成了数据库驱动的 JAR 包。



2. JDBC 包含两个部分：

(1) java.sql 包和 javax.sql 包中的 API。因为为了项目代码的可移植性，可维护性，SUN 公司在最初就制定了 Java 程序连接各种数据库的统一接口规范。这样的话，不管是连接哪一种 DBMS 软件，Java 代码可以保持一致性。

(2) 各个数据库厂商提供的 JAR 包。因为各个数据库厂商的 DBMS 软件各有不同，那么内部如何通过 sql 实现增、删、改、查等管理数据，只有这个数据库软件的厂商自己更清楚，因此把接口规范的实现交给各个数据库厂商自己实现。

3. 数据库连接池 (Database Connection Pool) 是一种管理数据库连接的技术。它在应用启动时创建一定数量的数据库连接，并将这些连接存储在一个连接池中。当应用程序需要与数据库交互时，它可以从连接池中获取一个可用的连接，而不是每次都创建一个新的连接。在完成数据库操作后，应用程序将连接返回到连接池中，以便其他应用程序或线程在需要时重复使用。

4. 数据库连接池的主要优点：

(1) 通过重用现有连接，减少了创建和关闭连接所需的时间，从而提高了应用程序的性能。

(2) 连接池限制了同时打开的连接数量，有助于防止系统资源耗尽。连接池还可以在空闲时自动关闭连接，以减少资源占用。

(3) 当应用程序的负载增加时，可以通过增加连接池的大小来提高处理能力。同样，当负载减少时，可以减小连接池的大小以节省资源。

5. Druid (德鲁伊) 数据库连接池：德鲁伊 (Druid) 数据库连接池是一个高性能、可扩展的 Java 数据库连接池，由阿里巴巴公司开发并开源。德鲁伊连接池旨在提供一个可靠的数据库连接池解决方案，具有高性能、强大的监控和扩展性，是目前业界主流的数据库连接池。

6. 使用德鲁伊连接池连接数据库的步骤：

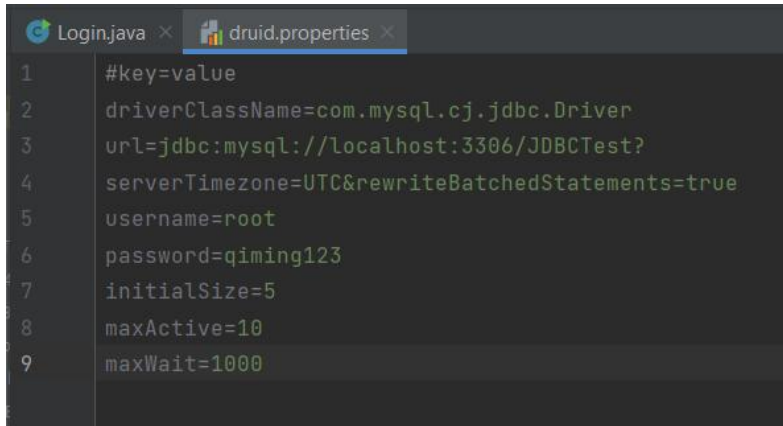
(1) 引入 jar 包

(2) 编写配置文件 (druid.properties 文件)

(3) 创建数据库连接池对象

(4) 获取连接

配置文件信息如下：



```
1 #key=value
2 driverClassName=com.mysql.cj.jdbc.Driver
3 url=jdbc:mysql://localhost:3306/JDBCTest?
4 serverTimezone=UTC&rewriteBatchedStatements=true
5 username=root
6 password=qiming123
7 initialSize=5
8 maxActive=10
9 maxWait=1000
```

7. 为什么使用 Java 开发数据库管理系统：

(1) Java 是一种跨平台的编程语言，通过 JVM（Java 虚拟机）技术，可在不同的操作系统（如 Windows、macOS、Linux）上运行。使用 Java 开发数据库管理系统，可以确保系统有良好的可移植性和平台独立性（一次编译，到处运行）。

(2) Java 具有一个庞大而极其成熟的生态系统，提供了大量的库和框架以支持开发各种应用程序，包括数据库管理系统。例如，Java 有 JDBC 库，可以简化与各种数据库的交互。另外 Java 和 MySQL、Oracle 数据库都是由 Oracle 公司运营和维护，因此版本越新，Java 本身对主流数据库的支持就越好。

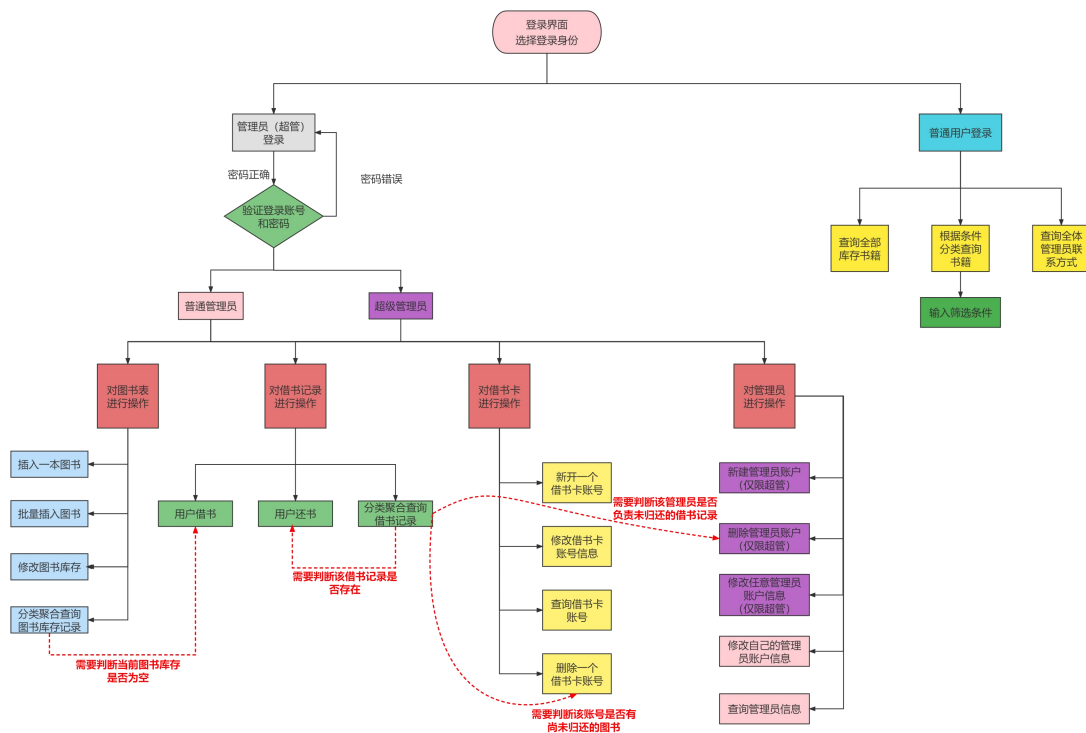
(3) Java 提供了良好的多线程支持，这对于数据库管理系统非常重要，因为它们需要同时处理来自多个客户端的请求。Java 的多线程支持可以帮助开发者实现高性能、高并发的数据库管理系统。

(4) Java 设计时考虑安全性，提供许多内置的安全特性，如沙箱模型、安全管理器和访问控制。可以构建安全的数据库管理系统，防止潜在的安全漏洞。

8. Swing: Swing 是 Java 的一个 GUI（图形用户界面）工具包，它是 Java Foundation Classes（Java 的基础类库）的一部分。Swing 提供了一组可重用的 GUI 组件，例如按钮、文本框、标签、表格、下拉列表、对话框等，使开发人员可以轻松地创建跨平台的 GUI 应用程序。Swing 还提供了一些高级组件，例如树形结构、表格和分隔窗格，可以更轻松地实现复杂而精美的 GUI 界面。

三、 操作方法与实验步骤

(1) 总流程图



(3) 具体实现方法:

1. 分类登录的实现方法:

- (1) 根据用户的选择进行处理，普通用户登录不需要密码
- (2) 管理员登录需要验证密码，将用户输入的账户和密码与数据库中的密码进行比对，判断是否成功登录。根据用户输入的账户自动判断是管理员还是超级管理员。（超级管理员可以对管理员账户进行增删改的操作）



2. 批量添加书籍的实现方法:

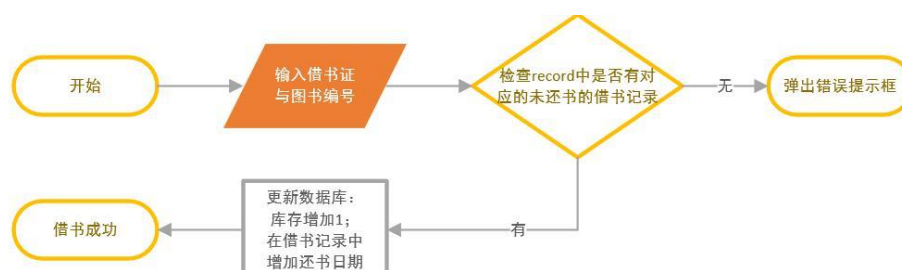
在 LabCode.Test 包中的 FileIOTest 类中有具体的实现。

通过读入文件的绝对路径，将文件内容存放在一个 List 中，通过正则表达式进行字符串的匹配，逐个添加到表中。

```
public static book parseBook(String line, int lineNumber){
    Pattern pattern =
    Pattern.compile("^\\s*(\\d+)\\s*,\\s*([^\"]+)\\s*,\\s*([^\"]+)\\s*,\\s*([^\"]+)\\s*,\\s*(\\d+)\\s*,\\s*"
    ,\\s*([^\"]+)\\s*,\\s*([^\"]+)\\s*,\\s*(\\d+)\\s*$");
    Matcher matcher = pattern.matcher(line);
    if (matcher.matches()) {
        int bno = Integer.parseInt(matcher.group(1));
        String category = matcher.group(2).trim();
        String title = matcher.group(3).trim();
        String press = matcher.group(4).trim();
        int year = Integer.parseInt(matcher.group(5));
        String author = matcher.group(6).trim().trim();
        BigDecimal price = new BigDecimal(matcher.group(7).trim());
        int total = Integer.parseInt(matcher.group(8));
        return new book(bno, category, title, press, year, author, price, total);
    }
    else{
        System.out.println("第 " + lineNumber + " 行格式错误: " + line);
        return null;
    }
}
```

3. 添加一条借书记录的方法

- (1) 首先需要判断该记录对应的书籍当前库存量是否大于 0，
- (2) 若库存量为 0，则不允许借出，
- (3) 若库存量大于 0，则添加一条借书记录，并使该书籍的库存量-1。



4. 删除借书卡的方法

- (1) 首先需要判断该借书卡对应的借书记录是否存在未归还的项，
- (2) 若有未归还的图书，则不允许删除借书卡，
- (3) 归还图书后，可以删除该卡，并把借书记录中的所有该卡的信息删除。

四、 实验结果与讨论、心得

在本次实验中，我使用 Java 标准类库中的 Swing GUI 工具，设计并实现了一个基于 MySQL 数据库的 Java 图书管理系统。该系统主要包括图书入库、查询、借书、还书、借书证管理等功能。通过本次实验，我提高了自己的系统编程能力，加深了对数据库系统原理及应用的理解。

在进行系统设计之前，我首先对所需的功能进行了详细的需求分析。根据需求分析，确定了以下五个核心功能：

- (1) 图书入库：支持管理员录入图书信息，包括书名、作者、出版社、分类、库存等信息。
- (2) 查询功能：方便读者和管理员检索图书信息，支持按书名、作者、分类等条件进行查询。

(3) 借书功能：允许读者借阅图书，并记录借阅信息，以便于管理员跟踪图书借阅状态。

(4) 还书功能：支持读者归还图书，并更新图书库存信息。

(5) 借书证管理：管理员可以管理读者的借书证，包括添加、删除、修改读者信息等操作。其中超级管理员还可以对普通管理员进行管理。

在数据库设计时，我根据系统需求分析，我设计了以下四个表存储相关信息：

(1) 图书信息表 (book)：存储图书的基本信息，包括书名、作者、出版社、分类、库存等。

(2) 读者信息表 (card)：记录读者的基本信息，如姓名、院系、类别、借书证号等。

(3) 借阅信息表 (borrow)：记录每次读者借阅图书的详细信息，包括借书证号、图书编号、借阅日期、归还日期等。

(4) 管理员信息表 (admin)：存储管理员的账号、密码等信息，用于登录和管理系统。

在系统架构上，为了实现一个具备良好扩展性和可维护性的图书管理系统，我采用了业界通用的 MVC (Model-View-Controller) 设计模式对系统进行了分层。系统主要分为以下三层：

模型层 (Model)：负责处理业务逻辑和数据操作，包括与数据库的交互等。

视图层 (View)：负责展示数据和接收用户输入，提供友好的用户界面。

控制层 (Controller)：负责接收视图层的请求，调用模型层的业务逻辑，并将处理结果返回给视图层。

在系统实现过程中，我采用了 Java 进行编程，并使用了以下技术和工具。

(1) MySQL 数据库：用于存储系统的数据信息。

(2) JDBC (Java Database Connectivity)：作为 Java 和 MySQL 数据库之间的桥梁，实现数据库操作的功能。

(3) Java Swing：用于创建图形用户界面 (GUI)。

通过以上技术和工具的应用，我成功地实现了图书管理系统的五个核心功能，满足了需求分析中的各项要求，设计并实现了一个基于 MySQL 数据库的 Java 图书管理系统，提高了自己的编程能力，加深了对数据库系统原理及应用的理解。

虽然目前的系统已经具备基本功能，但仍有许多可以优化和改进的地方。在未来的工作中，我希望将继续加强系统功能的完善，提高系统的易用性、稳定性和安全性，为用户提供更优质的服务。我还将关注数据库和人工智能新兴技术的发展趋势，引入更多先进技术，使系统具备更强大的功能和更高的性能。

五、 参考资料

《Java 核心技术 卷 I》Cay S. Horstmann

《Thinking in Java》Bruce Eckel

《Java 语言程序设计与数据结构》Daniel Y. Liang

《Java 语言程序设计（MOOC 版）》阎道宏

参考网站：

<https://docs.oracle.com/en/java/javase/19/docs/api/index.html>

<http://www.ultimategameprogramming.com>

<https://dev.mysql.com/doc/relnotes/connector-j/8.0/en/>

<https://dev.mysql.com/doc/connector-net/en/>