

# Database System实验五

完成了图书查询结果排序功能.

## 1系统整体架构

本项目使用Java语言开发， maven管理依赖。

### 工程结构

```
myLab5
├── .idea
├── databooks # 存储多批导入图书的数据
├── pom.xml # 确定包依赖
├── logs # log4j日志记录文件
├── src
│   ├── java/org.lab5
│   │   ├── entity # 存储对应数据库的实体类
│   │   │   ├── Book
│   │   │   ├── Card
│   │   │   ├── Admin # 管理员类
│   │   │   ├── queryCondition # 针对区间查询，继承Book类加入两个属性传入mybatis
│   │   │   └── Record # 借书记录
│   │   ├── admin # 管理员登陆后的一些方法
│   │   ├── Main # 程序入口
│   │   └── admin # 借书， 还书， 查询三个方法
│   └── resources
│       ├── bookMapper.xml # 操作book数据库的sql语句
│       ├── cardMapper.xml
│       ├── adminMapper.xml
│       ├── recordMapper.xml
│       ├── log4j.properties # log4j的配置文件
│       └── mybatis-config.xml # mybatis的配置文件
```

## 2程序设计中采用的技术

### jdbc

JDBC是Java DataBase Connectivity的缩写，它是Java程序访问数据库的标准接口。

使用Java程序访问数据库时，Java代码并不是直接通过TCP连接去访问数据库，而是通过JDBC接口来访问，而JDBC接口则通过JDBC驱动来实现真正对数据库的访问。

例如，我们在Java代码中如果要访问MySQL，那么必须编写代码操作JDBC接口。注意到JDBC接口是Java标准库自带的，所以可以直接编译。而具体的JDBC驱动是由数据库厂商提供的，例如，MySQL的JDBC驱动由Oracle提供。因此，访问某个具体的数据库，我们只需要引入该厂商提供的JDBC驱动，就可以通过JDBC接口来访问，这样保证了Java程序编写的是一套数据库访问代码，却可以访问各种不同的数据库，因为他们都提供了标准的JDBC驱动。

实际上，一个MySQL的JDBC的驱动就是一个jar包，它本身也是纯Java编写的。我们自己编写的代码只需要引用Java标准库提供的java.sql包下面的相关接口，由此再间接地通过MySQL驱动的jar包通过网络访问MySQL服务器，所有复杂的网络通讯都被封装到JDBC驱动中，因此，Java程序本身只需要引入一个MySQL驱动的jar包就可以正常访问MySQL服务器。

使用JDBC的好处是：

- 各数据库厂商使用相同的接口，Java代码不需要针对不同数据库分别开发；
- Java程序编译期仅依赖java.sql包，不依赖具体数据库的jar包；
- 可随时替换底层数据库，访问数据库的Java代码基本不变。

我们不直接写JDBC的resultset，而是用mybatis包装，可以直接操作实体类。

## mybatis

- 1、jdbc写查询返回resultset是不释放连接的，操作不好就容易出现死锁。mybatis框架的查询是自动释放连接的
- 2、jdbc防止SQL注入是比较麻烦的，mybatis防止SQL注入写起来比较简单的
- 3、jdbc写SQL会把SQL写到java代码中，如果在生产环境里，无法快速修改SQL，而mybatis的SQL在xml文件中，生产环境中可以修改xml后，重启即生效。
- 4、mybatis有默认缓存机制，对查询进行了缓存，jdbc缓存你还得自己实现。
- 5、最显而易见的益处是mybatis连接数据库编码更快。

MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生类型、接口和 Java 的 POJO（Plain Old Java Objects，普通老式 Java 对象）为数据库中的记录。

MyBatis 的查询结果集都是自动映射封装的，单行查询将数据库一条数据封装成对应的Java对象。多行查询，先将每一行封装成对象，再将每个对象添加到集合中，最后返回一个List集合对象。

## 范围查询

新增加了一个类叫做queryCondition.java，并让该类继承Books类,里面属性是Year2,Price2,并生成get、set方法。利用mybatis中的 `<where>` 和 `<if>` 标签，实现动态SQL不确定条件的查询。

## log4j

利用常用的SLF4j作为日志接口.输出日志，而不是用 `System.out.println()`，有以下几个好处：

1. 可以设置输出样式，避免自己每次都写 `"ERROR: " + var`；
2. 可以设置输出级别，禁止某些级别输出。例如，只输出错误日志；
3. 可以重定向到文件，这样可以在程序运行结束后查看日志；
4. 可以按包名控制日志级别，只输出某些包的日志；

## 3 数据库表结构的描述

Book( ISBN , Category, Title, Publisher, Year, Author, Price, Total, Inventory) 主键ISBN

LibCard (cid, name, unit, category) 主键卡号

Records(ISBN, cid, BorrowTime, Due, AID) 外键书号. 外键管理员id

Book

属性	数据类型	是否主键	是否null
ISBN	varchar(100)	主键	not null
Category	varchar(100)		
Title	varchar(100)		
Publisher	varchar(100)		
pubyear	int		
Author	Varchar(100)		
Price	double		
Total	int		not null
Inventory	int		not null

```
CREATE TABLE lab5.Book (  
  ISBN varchar(50) NOT NULL,  
  Category varchar(100) NULL,  
  Title varchar(100) NULL,  
  Publisher varchar(100) NULL,  
  pubYear INT NULL,  
  Author varchar(100) NULL,  
  Price INT NULL,  
  Total INT NOT NULL,  
  Inventory INT NOT NULL  
)  
ENGINE=InnoDB  
DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;  
ALTER TABLE lab5.Book ADD CONSTRAINT Book_PK PRIMARY KEY (ISBN);
```

LibCard

属性	数据类型	是否主键	是否null
cid	int	主键	not null
name	Varchar(100)		not null
unit	Varchar(100)		
category	Varchar(10)		not null

```
CREATE TABLE lab5.LibCard (
  cid INT auto_increment NOT NULL,
  name varchar(100) NOT NULL,
  unit varchar(100) NULL,
  category varchar(10) NOT NULL,
  CONSTRAINT LibCard_PK PRIMARY KEY (cid)
)
ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

## Administrator

属性	数据类型	是否主键	是否null
AID	int	主键	not null
password	Varchar(100)		not null
Name	Varchar(100)		
Contact	Varchar(100)		

```
CREATE TABLE lab5.Administrator (
  AID INT NOT NULL,
  password varchar(100) NOT NULL,
  Name varchar(100) NULL,
  Contact varchar(100) NULL
)
ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
ALTER TABLE lab5.Administrator ADD CONSTRAINT Administrator_PK PRIMARY KEY (AID);
```

## Records

属性	数据类型	是否外键
ISBN	Varchar(100)	外键
cid	int	外键
BorrowTime	DATE	
Due	DATE	
AID	int	外键

```
CREATE TABLE lab5.Records (  
  ISBN varchar(100) NOT NULL,  
  cid INT NOT NULL,  
  BorrowTime DATE NOT NULL,  
  Due DATE NULL,  
  AID INT NOT NULL  
)  
  
ENGINE=InnoDB  
DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;  
  
ALTER TABLE lab5.Records ADD CONSTRAINT Records_FK FOREIGN KEY (ISBN) REFERENCES  
lab5.Book(ISBN) ON DELETE CASCADE ON UPDATE CASCADE;  
ALTER TABLE lab5.Records ADD CONSTRAINT Records_FK_1 FOREIGN KEY (cid) REFERENCES  
lab5.LibCard(cid) ON DELETE CASCADE ON UPDATE CASCADE;  
ALTER TABLE lab5.Records ADD CONSTRAINT Records_FK_2 FOREIGN KEY (AID) REFERENCES  
lab5.Administrator(AID) ON DELETE CASCADE ON UPDATE CASCADE;
```

## 4 每个功能模块的演示

---

### 功能模块

模块名称	功能描述
管理员登陆	输入管理员ID，密码；登入系统 或 返回ID/密码 错误。
图书入库	1. 单本入库 2. 批量入库：图书信息存放在文件中，每条图书信息为一行。一行中的内容如下（书号，类别，书名，出版社，年份，作者，价格，数量） Note: 其中年份、数量是整数类型；价格是两位小数类型；其余为字符串类型 Sample: (book_no_1, CS, ComputerArchitecture, xxx, 2004, xxx, 90.00, 2)
图书查询	要求可以对书的类别，书名，出版社，年份(年份区间)，作者，价格(区间) 进行查询，支持同时查询多个条件。返回最多前五十条符合条件的图书信息。每条图书信息包括以下内容: ( 书号, 类别, 书名, 出版社, 年份, 作者, 价格, 总藏书量, 库存 ); 可选要求: 可以按用户指定属性对图书信息进行排序. (默认是书名)
借书	1.输入借书证卡号 显示该借书证所有已借书籍 (返回, 格式同查询模块) 2.输入书号：如果该书还有库存，则借书成功，同时库存数减一。否则输出该书无库存，且输出预计最快的归还时间。
还书	1.输入借书证卡号：显示该借书证所有已借书籍 (返回, 格式同查询模块) 2.输入书号：如果该书在已借书籍列表内，则还书成功，同时库存加一。否则输出出错信息。
借书证管理	增加或删除一个借书证。

管理员登陆

输入管理员ID，密码；登入系统

```
/Users/juyilin/Library/Java/JavaVirtualMachines/op  
please enter 1- 4  
1 admin login  
2 query book  
3 borrow book  
4 return book  
1  
please enter admin id  
1  
please enter admin password  
12  
login successful!
```

返回ID/密码 错误。

```
↑ /Users/juyilin/Library/Java/JavaVirtualMachines/op  
↓  
⌵  
⌵  
⌵  
⌵  
↑ please enter 1- 4  
1 admin login  
2 query book  
3 borrow book  
4 return book  
1  
please enter admin id  
1  
please enter admin password  
13  
ID/password error!
```

图书入库

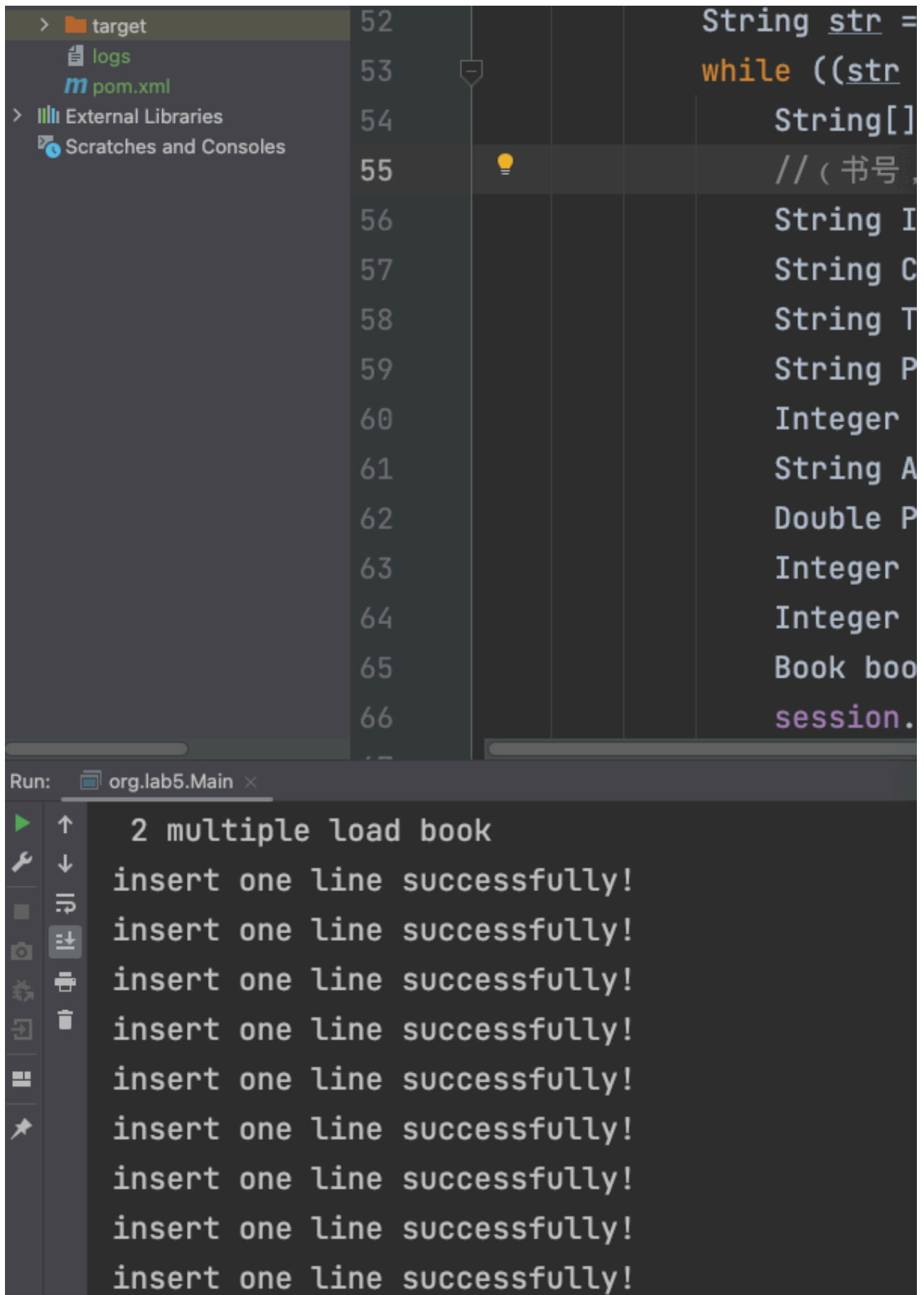
单行入库

```
please enter 1- 4 input 9 will exit
1 admin login
2 query book
3 borrow book
4 return book
please enter 1- 2 . input 9 will exit
1 single load book
2 multiple load book
book25,      EE, CS25,  PUB25,  2027,  author25,  91.11,  25
insert one line successfully!
```

网格		ISBN	Category	Title	Publisher	Author	Price	Total	Ir
	1	book26	EE	CS26	PUB26	author26	103	26	

批量入库





ISBN	Category	Title	Publisher	Author	Price	To
book_no_1	CS	ComputerArchitecture	xxx	xxx	90	
book_no_2	CS	ComputerNet	xxx	xxx	91	
book10	CS	CS10	PUB10	author10	95	
book11	CS	CS11	PUB11	author11	91.04	
book12	CS	CS12	PUB12	author12	96	
book13	CS	CS13	PUB13	author13	91.05	
book14	EE	CS14	PUB14	author14	97	
book15	EE	CS15	PUB15	author15	91.06	
book16	EE	CS16	PUB16	author16	98	
book17	EE	CS17	PUB17	author17	91.07	
book18	EE	CS18	PUB18	author18	99	
book19	EE	CS19	PUB19	author19	91.08	
book20	EE	CS20	PUB20	author20	100	
book21	EE	CS21	PUB21	author21	91.09	
book22	EE	CS22	PUB22	author22	101	
book23	EE	CS23	PUB23	author23	91.1	
book24	EE	CS24	PUB24	author24	102	
book25	EE	CS25	PUB25	author25	91.11	
book26	EE	CS26	PUB26	author26	103	
book3	CS	CS3	PUB3	author3	91	
book4	CS	CS4	PUB4	author4	92	
book5	CS	CS5	PUB5	author5	91.01	
book6	CS	CS6	PUB6	author6	93	
book7	CS	CS7	PUB7	author7	91.02	
book8	CS	CS8	PUB8	author8	94	
book9	CS	CS9	PUB9	author9	91.03	

## 图书查询

要求可以对书的类别，书名，出版社，年份(年份区间)，作者，价格(区间) 进行查询，支持同时查询多个条件。返回最多前五十条符合条件的图书信息。每条图书信息包括以下内容: ( 书号, 类别, 书名, 出版社, 年份, 作者, 价格, 总藏书量, 库存 ); 可选要求: 可以按用户指定属性对图书信息进行排序. (默认是书名)

如图, 同时查询多个条件,搜索EE, 2021-2028年, 价格105元以下的书.

最多前五十条 : sql语句的最后加上limit 50.

范围查询的传入样例

```

null,EE,null,null,2021,2028,null,91.0,91.5
book18,EE,CS18,PUB18, 2011,2028, author18,91.0,99.2
,EE, , ,2021,2028, , ,105

```

## 排序

可以根据用户指定属性对图书查询结果进行排序

实现方式： switch case对应各个属性, java 按类的指定属性排序.

默认按书名排序:

```
.EE, , ,2021,2028, , ,105
[ISBN : book20 Category :EE Title:CS20 Publisher: PUB20 Year : null Author: author20 Price : 100.0 Total : 20 Inventory: 20
[ISBN : book21 Category :EE Title:CS21 Publisher: PUB21 Year : null Author: author21 Price : 91.09 Total : 21 Inventory: 21
[ISBN : book22 Category :EE Title:CS22 Publisher: PUB22 Year : null Author: author22 Price : 101.0 Total : 22 Inventory: 22
[ISBN : book23 Category :EE Title:CS23 Publisher: PUB23 Year : null Author: author23 Price : 91.1 Total : 23 Inventory: 23
[ISBN : book24 Category :EE Title:CS24 Publisher: PUB24 Year : null Author: author24 Price : 102.0 Total : 24 Inventory: 24
[ISBN : book25 Category :EE Title:CS25 Publisher: PUB25 Year : null Author: author25 Price : 91.11 Total : 25 Inventory: 25
```

用户输入Price, 即可按价格排序.

```
Price
.EE, , ,2021,2028, , ,105
[ISBN : book21 Category :EE Title:CS21 Publisher: PUB21 Year : null Author: author21 Price : 91.09 Total : 21 Inventory: 21
[ISBN : book23 Category :EE Title:CS23 Publisher: PUB23 Year : null Author: author23 Price : 91.1 Total : 23 Inventory: 23
[ISBN : book25 Category :EE Title:CS25 Publisher: PUB25 Year : null Author: author25 Price : 91.11 Total : 25 Inventory: 25
[ISBN : book20 Category :EE Title:CS20 Publisher: PUB20 Year : null Author: author20 Price : 100.0 Total : 20 Inventory: 20
[ISBN : book22 Category :EE Title:CS22 Publisher: PUB22 Year : null Author: author22 Price : 101.0 Total : 22 Inventory: 22
[ISBN : book24 Category :EE Title:CS24 Publisher: PUB24 Year : null Author: author24 Price : 102.0 Total : 24 Inventory: 24
```

用mysql的实现方式： bookMapper.xml, 传入的queryCondition类 再多一个属性, sql语句最后加上 orderby即可.

## 借书

1.输入借书证卡号 显示该借书证所有已借书籍 (返回, 格式同查询模块) 2.输入书号：如果该书还有库存，则借书成功，同时库存数减一。否则输出该书无库存，且输出预计最快的归还时间。

显示借书证2所有已借书籍

```
[main] INFO org.lab5.normalUser - please input card id:
3
[main] INFO org.lab5.normalUser - showing all books...
[ISBN : book_no_2 Category :CS Title:ComputerNet Publisher: xxx Year : null Author: xxx Price : 91.0 Total : 3 Inventory: 2
[ISBN : book10 Category :CS Title:CS10 Publisher: PUB10 Year : null Author: author10 Price : 95.0 Total : 10 Inventory: 9
```

借书证1 借书成功

```
[main] INFO org.lab5.normalUser - please input isdn:
book_no_1
[main] INFO org.lab5.normalUser - searching books...
[main] INFO org.lab5.normalUser - borrow book success!
```

lab5

Administrator

Records X

Book

LibCard

属性

数据

ER 图

Records

输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

网格

文本

	ISBN	cid	BorrowTime	Due	AID	
1	book_no_1	1	2022-04-25	2022-06-04	1	

Book 输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)									
	ISBN	ABC Category	ABC Title	ABC Publisher	ABC Author	123 Price	123 Total	123 Inventory	123 pubyear
1	book_no_1	CS	ComputerArchitecture	xxx	xxx	90	2	1	2,004
2	book_no_2	CS	ComputerNet	xxx	xxx	91	3	3	2,005
3	book10	CS	CS10	PUB10	author10	95	10	10	2,012
4	book11	CS	CS11	PUB11	author11	91.04	11	11	2,013
5	book12	CS	CS12	PUB12	author12	96	12	12	2,014
6	book13	CS	CS13	PUB13	author13	91.05	13	13	2,015
7	book14	EE	CS14	PUB14	author14	97	14	14	2,016
8	book15	EE	CS15	PUB15	author15	91.06	15	15	2,017
9	book16	EE	CS16	PUB16	author16	98	16	16	2,018
10	book17	EE	CS17	PUB17	author17	91.07	17	17	2,019

可以看到book\_no\_1的inventory从2 变成了1.

借书失败

2 BORROW BOOK

```

2
[main] INFO org.lab5.normalUser - please input isdn:
book_no_2
[main] INFO org.lab5.normalUser - searching books...
[main] ERROR org.lab5.normalUser - This book doesn't have inventory!
[main] INFO org.lab5.normalUser - The nearest due is : 2022-06-04

```



2

```
[main] INFO org.lab5.normalUser - please input isdn:
```

book10

```
[main] INFO org.lab5.normalUser - searching books...
```

```
[main] INFO org.lab5.normalUser - return book success!
```

	ABC ISBN	ABC Category	ABC Title	ABC Publisher	ABC Author	123 Price	123 Total	123 Inventory	123 pubyear
1	book_no_1	CS	ComputerArchitecture	xxx	xxx	90	2	0	2,004
2	book_no_2	CS	ComputerNet	xxx	xxx	91	3	0	2,005
3	book10	CS	CS10	PUB10	author10	95	10	10	2,012
4	book11	CS	CS11	PUB11	author11	91.04	11	11	2,013
5	book12	CS	CS12	PUB12	author12	96	12	12	2,014
6	book13	CS	CS13	PUB13	author13	91.05	13	13	2,015
7	book14	EE	CS14	PUB14	author14	97	14	14	2,016
8	book15	EE	CS15	PUB15	author15	91.06	15	15	2,017
9	book16	EE	CS16	PUB16	author16	98	16	16	2,018

还书成功，同时库存加一

card2归还book11

```
1 show books
2 return book
```

2

```
[main] INFO org.lab5.normalUser - please input isdn:
```

book11

```
[main] INFO org.lab5.normalUser - searching books...
```

```
[main] ERROR org.lab5.normalUser - This book isn't borrowed!
```

输出出错信息

借书证管理

加入借书证

```
2 multiple load book
```

```
3 add Libcard
```





```
4 remove Libcard
```

```
input new card info
```

```
1, card1, unit1, category1
```

```
[main] INFO org.lab5.admin - insert card successfully!
```

oCard |  输入一个 SQL 表达式来过滤结果 (使用 *Ctrl+Space*)

	123 cid 	ABC name 	ABC unit 	ABC category 
	1	card1	unit1	category1

删除借书证

```
2 query book
3 borrow book
4 return book

please enter 1- 2 . input 9 will e

1 single load book
2 multiple load book
3 add Libcard
4 remove Libcard

input cid

1

delete card successfully!
```

[illegible]