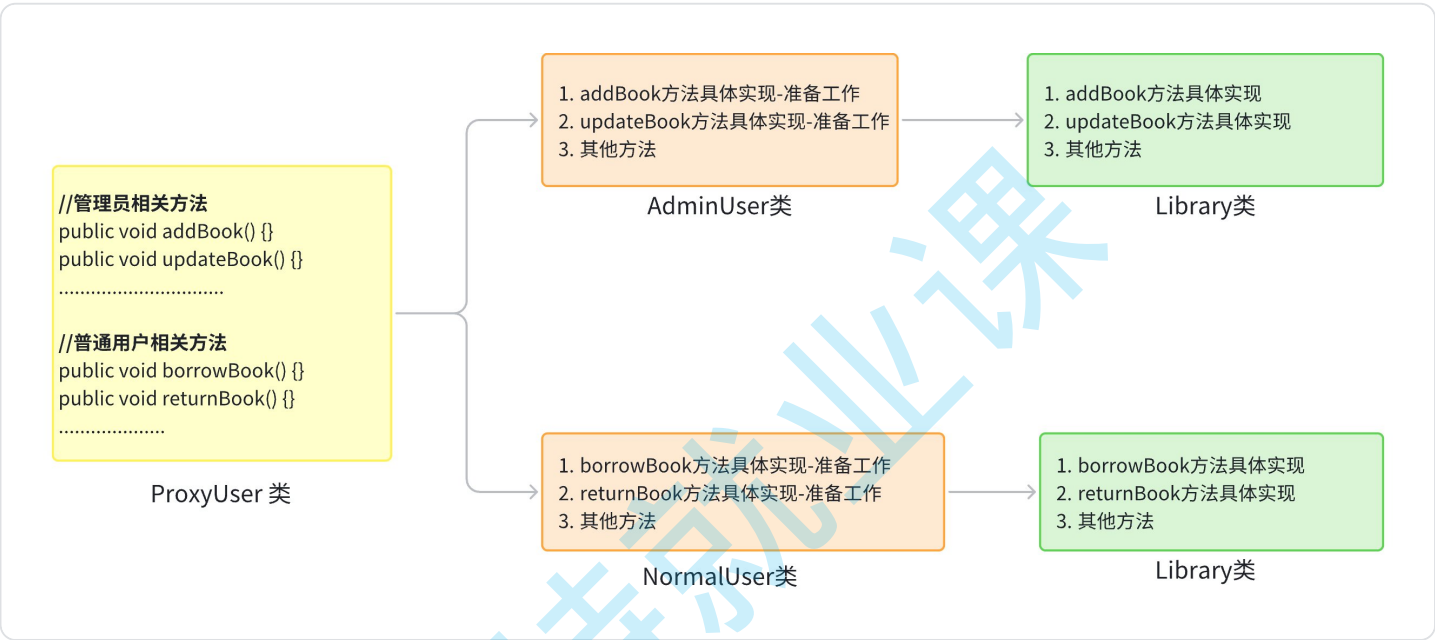


22.图书系统项目（六）

本节主要完成普通用户业务逻辑具体实现

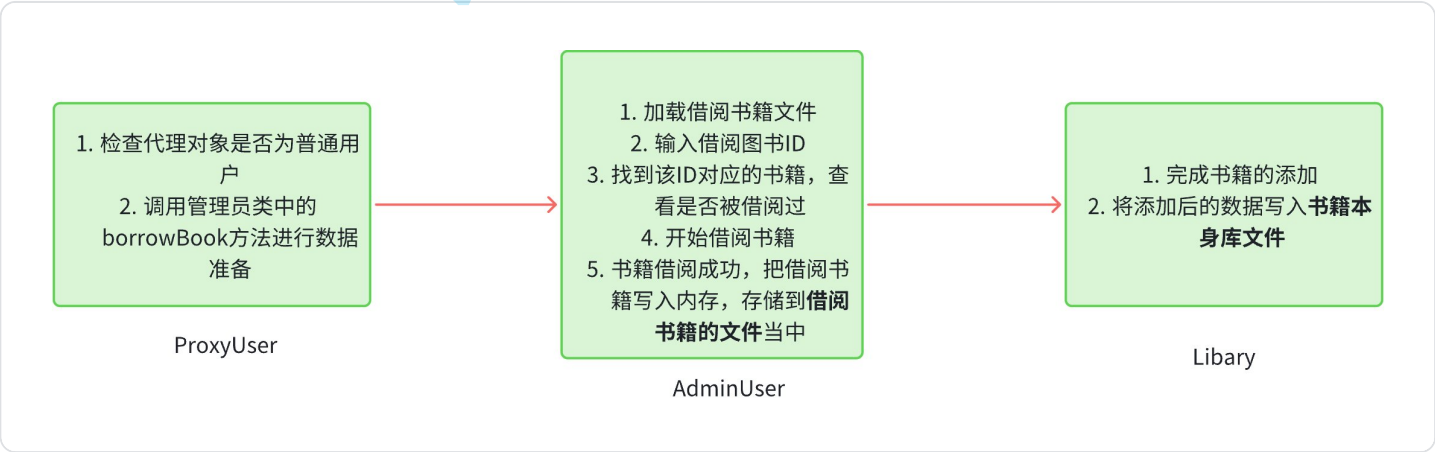
1. 回顾业务流程

我们前面已经分析过了，整个业务逻辑框架的实现流程如下：



2. 普通用户端业务实现

2.1 借阅图书



1. ProxyUser类：

```

2  * 检查当前代理用户是否为普通用户
3  */
4  private void checkRealUserWhetherNormalUser(String exceptionMessage) {
5      if(!(realUser instanceof NormalUser)){
6          throw new PermissionException(exceptionMessage);
7      }
8  }
9  //借阅图书
10 public void borrowBook() {
11     checkRealUserWhetherNormalUser("管理员请以普通用户的方式借阅图书");
12     ((NormalUser) realUser).borrowBook();
13 }

```

2. NormalUser类:

在借阅书籍文档当中存储数据的格式约定为：userID,bookId 通过换行的方式，存储多个用户的借阅记录。

2,3
2,1

每一行数据，使用类 `PairOfUidAndBookId` 来进行存储

```

1  public class PairOfUidAndBookId {
2      private int userId;
3
4      private int bookId;
5
6      public PairOfUidAndBookId() {
7
8      }
9
10     public PairOfUidAndBookId(int userId, int bookId) {
11         this.userId = userId;
12         this.bookId = bookId;
13     }
14
15     public int getUserId() {
16         return userId;
17     }
18
19     public void setUserId(int userId) {
20         this.userId = userId;
21     }

```

```

22
23     public int getBookId() {
24         return bookId;
25     }
26
27     public void setBookId(int bookId) {
28         this.bookId = bookId;
29     }
30
31     //存储书籍是进行转换字符串
32     public String toJson() {
33         StringBuilder json = new StringBuilder();
34         json.append(userId).append(",");
35         json.append(bookId);
36         return json.toString();
37     }
38 }

```

方法实现：

```

1  //借阅图书
2  public void borrowBook() {
3      scanner.nextLine();
4      System.out.println("请输入你要借阅图书的id: ");
5      int bookId = scanner.nextInt();
6      scanner.nextLine();
7
8      //如果书架没有书 不能借阅
9      if (library.getBookCount() == 0) {
10         System.out.println("书架没有书籍暂且不能借阅");
11         return;
12     }
13     //加载已借阅的图书信息
14     loadBorrowedBook();
15
16     //判断要借阅的书 是否是已经被借阅过了
17     Book book = library.searchById(bookId);
18     if(book == null) {
19         System.out.println("没有该id的相关书籍: "+bookId);
20         return;
21     }
22     //检查通过loadBorrowedBook()加载到pairOfUidAndBookIds数组当中
23     //是否有bookId == 当前需要借阅的图书ID && UID也是一样的 说明当前用户借阅过
24     //否则就是其他用户借阅过
25     for (int i = 0; i < borrowedCount; i++) {

```

```

26         PairOfUidAndBookId pairOfUidAndBookId = pairOfUidAndBookIds[i];
27         //找到了对应的书籍
28         if (pairOfUidAndBookId.getBookId() == book.getBookId()) {
29             if (getUserID() == pairOfUidAndBookId.getUserId()) {
30                 System.out.println("该书已经被你借阅过了，你的ID是：" +
getUserID());
31                 return;
32             } else {
33                 System.out.println("该书已经被其他人借阅过了，他的ID是：" +
pairOfUidAndBookId.getUserId());
34                 return;
35             }
36         }
37     }
38
39     library.borrowBook(bookId);
40
41     //封装对象写到 借阅表中
42     PairOfUidAndBookId pairOfUidAndBookId = new PairOfUidAndBookId(userID,
book.getBookId());
43     pairOfUidAndBookIds[borrowedCount] = pairOfUidAndBookId;
44     borrowedCount++;
45
46     //存储借阅图书
47     storeBorrowedBook();
48
49     System.out.println("借阅成功！");
50 }

```

3. Library类

```

1  //借阅图书
2  public void borrowBook(int bookId) {
3      loadAllBook();
4      for (int i = 0; i < getBookCount(); i++) {
5          Book book = books[i];
6          if (book.getBookId() == bookId) {
7              book.setBorrowed(true);
8              book.incrementBorrowCount();
9          }
10     }
11     storeBook();
12 }
13
14 Book类新增方法：

```

```

15 public void incrementBorrowCount() {
16     this.borrowCount++;
17 }
18
19 public void decreaseBorrowCount() {
20     this.borrowCount--;
21 }

```

2.2 归还图书

1. ProxyUser类:

```

1 //归还图书
2 public void returnBook() {
3     checkRealUserWhetherNormalUser("管理员请以普通用户的方式归还图书");
4     ((NormalUser) realUser).returnBook();
5 }

```

2. NormalUser类:

- 归还图书需要注意，内存中的数组 `pairOfUidAndBookIds` 也需要删除，同时有效记录数 `borrowedCount` 也要相应改变
- 对借阅书籍文件和全部书籍文件都要进行更新操作

```

1 //归还图书
2 public void returnBook() {
3     loadBorrowedBook();
4
5     if (borrowedCount == 0) {
6         System.out.println("目前没有用户借阅过书籍");
7         return;
8     }
9     scanner.nextLine();
10    System.out.println("请输入你要归还图书的id: ");
11    int bookId = scanner.nextInt();
12    scanner.nextLine();
13
14    //判断要借阅的书 是否是已经被自己借阅过了
15    Book book = library.searchById(bookId);
16    if (book == null) {
17        System.out.println("没有该id的相关书籍: "+bookId);

```

```

18         return;
19     }
20     for (int i = 0; i < borrowedCount; i++) {
21         //如果2本书的ID, 则认为是同一本书
22         if (pairOfUidAndBookIds[i].getBookId()==book.getBookId()) {
23             //借阅用户也一样
24             if (getUserID() == pairOfUidAndBookIds[i].getUserId()) {
25                 library.returnBook(bookId);
26                 System.out.println("图书 '" + book.getTitle() + "' 已成功归
还。");
27                 // 用最后一本替换归还的书
28                 pairOfUidAndBookIds[i] = pairOfUidAndBookIds[borrowedCount -
1];
29                 // 清空最后一个
30                 pairOfUidAndBookIds[borrowedCount - 1] = null;
31                 borrowedCount--;
32                 storeBorrowedBook();
33             }else{
34                 System.out.println("该书籍不是你借阅的书籍, 不能归
还: "+book.getTitle());
35             }
36             return;
37         }
38     }
39     System.out.println("你没有借阅该书籍, 不需要归还, 书籍ID为: "+bookId);
40 }

```

3. Library类

```

1 //归还图书
2 public void returnBook(int bookId) {
3     loadAllBook();
4     for (int i = 0; i < getBookCount(); i++) {
5         Book book = books[i];
6         if(book.getBookId()==bookId) {
7             book.setBorrowed(false);
8             book.decreaseBorrowCount();
9         }
10    }
11    storeBook();
12 }

```

2.3 查看个人借阅情况

1. ProxyUser类:

```
1  //查看个人借阅情况
2  public void viewBorrowHistory() {
3      checkRealUserWhetherNormalUser("管理员请以普通用户的方式查看个人当前借阅情况");
4      ((NormalUser) realUser).viewBorrowBooks();
5  }
```

2. NormalUser类:

- 读取数据到内存数组
- 遍历内存数组即可

```
1  // 查看个人借阅情况
2  public void viewBorrowBooks() {
3      //读取当前借阅所有用户的借阅书籍的情况
4      loadBorrowedBook();
5      System.out.println("您的借阅情况如下: ");
6      if (borrowedCount == 0) {
7          System.out.println("目前没有借阅记录.....");
8      } else {
9          boolean flg = false;
10         for (int i = 0; i < borrowedCount; i++) {
11             //这里只能查看属于自己借阅的情况
12             //用户ID相同的情况下, 使用书籍ID查询书籍
13             if(pairOfUidAndBookIds[i].getUserId() == userID) {
14                 flg = true;
15                 Book book =
16                     library.searchById(pairOfUidAndBookIds[i].getBookId());
17                 System.out.println(book);
18             }
19             if(!flg) {
20                 System.out.println("你没有借阅过书籍! ");
21             }
22         }
23     }
24 }
```

3. Library类

1 不进行实现

3. 共用方法实现

以下方法为普通用户和管理员用户共用方法，这些方法存放在 `Library` 类中实现

3.1 查找图书

```
1  //根据书名进行查找 - 该方法是 管理员和普通用户都要进行访问，不在
2  public void searchBook() {
3      scanner.nextLine();
4      System.out.println("请输入你要查找的图书的名称：");
5      String title = scanner.nextLine();
6
7      Book book = search(title);
8      if(book == null) {
9          System.out.println("没有你要找的这本书，你查找的书名为："+title);
10     }else {
11         System.out.println("找到了你要查找的书，书的详细信息如下：");
12         System.out.println(book);
13     }
14 }
15
16 //根据title查找对应书籍
17 private Book search(String title) {
18     loadAllBook();
19     for (int i = 0; i < getBookCount(); i++) {
20         Book book = books[i];
21         if(book.getTitle().equals(title)) {
22             return book;
23         }
24     }
25     return null;
26 }
```

3.2 显示图书

1 //打印所有图书


```
2 public void displayBooks() {
3     System.out.println("当前图书馆当中所有的图书: ");
4     //更新一次文件当中的新数据
5     loadAllBook();
6
7     for (int i = 0; i < bookCount; i++) {
8         Book book = books[i];
9         System.out.println(book);
10    }
11 }
```

3.3 退出系统

```
1 //退出系统
2 public void exit() {
3     System.out.println("退出系统! ");
4     System.exit(0);
5 }
```

4. 项目最后

该项目已经完成了具体的实现过程，包括框架搭建，业务实现

- 整体运行一遍项目，进行各个功能的测试
- 作为扩展可以把项目的业务再进行丰富
- 后续学习完成数据结构，数据库之后，可以对该项目进行进一步更新迭代
- 先听会，再写会，最后讲出来。

完