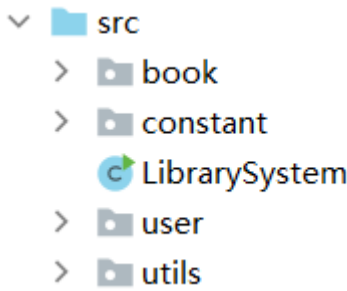


# 18.图书系统项目（二）

## 1. 模块划分介绍

图书系统中模块的划分共划分为4块：用户模块、书籍相关模块、工具相关模块、常量值相关模块。我们依次将4个模块在IDEA当中进行创建，供后续使用。



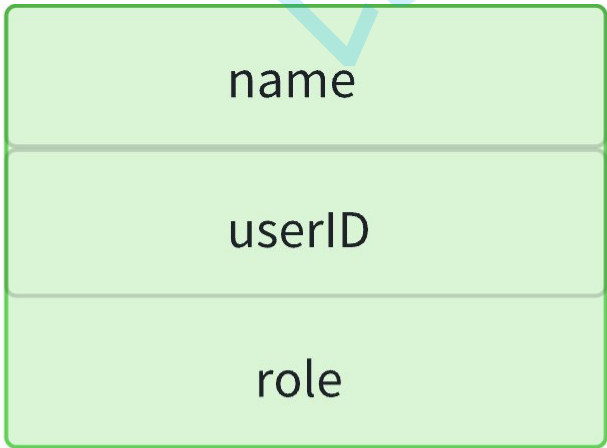
其中LibrarySystem模块为程序的入口。

## 2. User模块

从角色上我们将图书系统划分为管理员模块和普通用户模块，同时我们支持多普通用户进行登录。每个用户的属性包含：姓名、ID、角色3个属性。

### 2.1 User父类对象

相比较上一小节的user类，多了一个角色的属性



user对象

考虑到后续普通用户和管理员都有对应的菜单，我们将菜单方法也进行同步处理。代码如下：

```

1  public abstract class User {
2      protected String name;    // 用户名
3      protected int userID;     // 用户ID
4      protected String role;    // 用户角色 (管理员或普通用户)
5
6      // 构造函数
7      public User(String name, int userID, String role) {
8          this.name = name;
9          this.userID = userID;
10         this.role = role;
11     }
12
13     public String getName() {
14         return name;
15     }
16
17     public int getUserID() {
18         return userID;
19     }
20
21     public String getRole() {
22         return role;
23     }
24
25     /**
26      * 定义菜单方法 该方法打印菜单同时 输出菜单的选择
27      * @return
28      */
29     public abstract int display();
30 }

```

## 2.2 AdminUser对象

```

1  public class AdminUser extends User{
2      public AdminUser(String name,int userID) {
3          super(name,userID,"管理员");
4      }
5
6      @Override
7      public int display() {
8          System.out.println("管理员 " + name + " 的操作菜单:");
9          System.out.println("1. 查找图书");
10         System.out.println("2. 打印所有的图书");
11         System.out.println("3. 退出系统");
12         System.out.println("4. 上架图书");

```

```
13         System.out.println("5. 修改图书");
14         System.out.println("6. 下架图书");
15         System.out.println("7. 统计借阅次数");
16         System.out.println("8. 查看最后欢迎的前K本书");
17         System.out.println("9. 查看库存状态");
18         System.out.println("10. 按类别统计图书 ");
19         System.out.println("11. 按作者统计图书 ");
20         System.out.println("12. 检查超过一年未下架的图书");
21         System.out.println("请选择你的操作: ");
22         return scanner.nextInt();
23     }
24     //其他操作方法
25     //上架图书
26     public void addBook() {
27
28     }
29
30     //图书修改 支持修改书名 作者 类别
31     public void updateBook() {
32
33     }
34
35     //删除书籍
36     public void removeBook() {
37
38     }
39
40     //统计每本书的借阅次数
41     public void borrowCount() {
42
43     }
44
45     //查询最受欢迎的前n本书
46     public void generateBook() {
47
48     }
49
50     //查看库存状态
51     public void checkInventoryStatus() {
52
53     }
54
55     //按照类别 统计图书
56     public void categorizeBooksByCategory() {
57
58     }
59
```

```

60      //按照作者统计图书
61      public void categorizeBooksByAuthor() {
62
63      }
64
65      //并移除上架超过一年的图书
66      public void checkAndRemoveOldBooks() {
67
68      }
69
70      public void exit() {
71
72      }
73  }

```

## 2.3 NormalUser对象

```

1  public class NormalUser extends User {
2
3      //如果是普通用户，这里写死
4      public NormalUser(String name, int userID) {
5          super(name, userID, "普通用户");
6
7      }
8
9      private void loadBorrowedBook() {
10
11     }
12
13     private void storeBorrowedBook() {
14
15     }
16
17     @Override
18     public int display() {
19         System.out.println("普通用户 " + name + " 的操作菜单:");
20         System.out.println("1. 查找图书");
21         System.out.println("2. 打印所有的图书");
22         System.out.println("3. 退出系统");
23         System.out.println("4. 借阅图书");
24         System.out.println("5. 归还图书");
25         System.out.println("6. 查看当前个人借阅情况");
26         System.out.println("请选择你的操作: ");
27         return scanner.nextInt();
28     }

```

```

29
30     //借阅图书
31     public void borrowBook() {
32
33     }
34
35
36     //归还图书
37     public void returnBook() {
38
39     }
40
41     // 查看个人借阅情况
42     public void viewBorrowBooks() {
43
44     }
45
46 }

```

## 2.4 LibrarySystem中整合当前用户相关信息

LibrarySystem类中每次通过new关键字来创建对象，如果有多个普通用户这里需要同样的代码new很多次，代码会有一些重复。

```

1  public class LibrarySystem{
2      public static void main(String[] args) {
3          // 直接创建管理员用户
4          User adminUser = new AdminUser("刘备", 1);
5          // 直接创建普通用户
6          User normalUser1 = new NormalUser("关羽", 2);
7
8          User normalUser2 = new NormalUser("张飞", 3);
9      }
10 }

```

## 3. User模块-引入工程方法模式

创建factory包

### 1. 创建工厂接口

```

1  public interface IUserFactory {

```

```
2     User createUser(String name, int userID);
3 }
```

## 2. 创建具体的工程

```
1  public class AdminUserFactory implements IUserFactory{
2      @Override
3      public User createUser(String name, int userID) {
4          return new AdminUser(name,userID);
5      }
6  }
7
8  public class NormalUserFactory implements IUserFactory{
9      @Override
10     public User createUser(String name, int userID) {
11         return new NormalUser(name,userID);
12     }
13 }
```

## 3. 修改 `LibrarySystem` 类

```
1  public class LibrarySystem{
2      public static void main(String[] args) {
3
4
5          IUserFactory adminUserFactory = new AdminUserFactory();
6          User adminUser = adminUserFactory.createUser("刘备",1);
7
8          IUserFactory normalUserFactory = new NormalUserFactory();
9          User normalUser1 = normalUserFactory.createUser("关羽",2);
10         User normalUser2 = normalUserFactory.createUser("张飞",3);
11
12     }
13 }
```

## 4. User模块-引入代理模式控制对象权限

通过代理真实的用户 `realUser` 通过构造方法进行传递。每一个操作通过权限检查进行调用。

```
1  public class ProxyUser {
2
```

```
3      //被代理的真实用户
4      private User realUser;
5
6      public ProxyUser(User user) {
7          //对被代理的对象进行赋值
8          realUser = user;
9      }
10
11     public User getRealUser() {
12         return realUser;
13     }
14
15
16
17     //其他操作
18     //调用菜单
19     public int display() {
20
21     }
22     //添加书籍操作
23     public void addBook() {
24
25     }
26
27     //更新书籍操作
28     public void updateBook() {
29
30     }
31
32     //移除图书
33     public void removeBook() {
34
35     }
36
37     //查看图书的借阅次数
38     public void borrowCount( ) {
39
40     }
41
42     //查看最受欢迎的前K本书
43     public void generateBook() {
44
45     }
46
47     //查看库存状态
48     public void checkInventoryStatus() {
49
```

```

50     }
51     //按照类别 统计图书
52     public void categorizeBooksByCategory() {
53
54     }
55
56     //按照作者 统计图书
57     public void categorizeBooksByAuthor() {
58
59     }
60
61     //移除上架超过1年的书籍
62     public void checkAndRemoveOldBooks() {
63
64     }
65
66     //-----普通相关方法-----
-//
67     //借阅图书
68     public void borrowBook() {
69
70     }
71
72     //归还图书
73     public void returnBook() {
74
75     }
76
77
78     //查看个人借阅情况
79     public void viewBorrowHistory() {
80
81     }
82 }

```

### LibrarySystem 类中进行整合

```

1  public class LibrarySystem{
2      public static void main(String[] args) {
3
4          IUserFactory adminUserFactory = new AdminUserFactory();
5          User adminUser = adminUserFactory.createUser("刘备",1);
6
7          IUserFactory normalUserFactory = new NormalUserFactory();
8          User normalUser1 = normalUserFactory.createUser("关羽",2);

```



```
9      User normalUser2 = normalUserFactory.createUser("张飞",3);
10      /**
11      * 1.4 使用代理模式来管理权限
12      * 使用代理模式来控制 对象的访问
13      */
14
15      ProxyUser proxyUserAdmin = new ProxyUser(adminUser);
16      ProxyUser proxyUserNormalG = new ProxyUser(normalUser1);
17      ProxyUser proxyUserNormalZ = new ProxyUser(normalUser2);
18
19    }
20 }
```

完

比特就业课