


分布式计算第三次作业

姓名：张泽群 学号：19049100002 课程号：CS205105

 西安电子科技大学
XIDIAN UNIVERSITY

计算机科学与技术学院
School of Computer Science and Technology

第三次作业！

1. 具体要求

利用RPC机制进行协作，服务器至少暴露以下端口。(本实验运用 JAVA RMI 中间件完成。)

```
boolean add(Book b)
Book queryByID(int bookID)
BookList queryByName(String name)
boolean delete(int bookID)
```

2. BookManager System 项目结构

```
BookManager
├── Book.java
├── BookList.java
├── MyInterface.java
├── Query.txt
├── BookManager.txt
├──
├── BookCilent
│   └── BookClient.java
├──
└── BookServer
    ├── BookMethodImpl.java
    └── BookServer.java
```

项目的根目录 `BookManager` 是项目名，其名下包括书籍类和书籍列表类 `Book.java` `BookList.java`，以及服务器端和客户端功能函数的公共接口 `MyInterface.java`，以及存放图书信息和查询信息的 `Query.txt` 和 `BookManager.txt`。

存放客户端程序源码的目录是 `BookManager/BookCilent`，存放服务器端程序源码的目录是 `src/main/resources`，存放测试源码的目录是 `src/test/java`，存放测试资源的目录是 `BookManager/BookServer`，其包括公共接口的具体实现 `BookMethodImpl.java`。

3. 功能函数列表

Book 类 功能函数

```
public int getID()           // 获取书籍ID
public void setID(int bookID) // 设置书籍ID
public String getAuthor()    // 获取作者名
public String getBookName()  // 获取书籍名称
public int getBookNum()      // 获取书籍库存量
public int addBookNum(int i)  // 设置书籍库存量=原书籍库存量 + i
public void setBookNum(int i) // 设置书籍库存量= i
public void showInfo()       // 展示书本信息
```

BookList 类 功能函数

```
public String getInfo() // 获取书籍列表中所有书籍的信息
public void showInfo()  // 显示书籍列表中所有书籍的信息
```

公共接口 MyInterface 功能函数

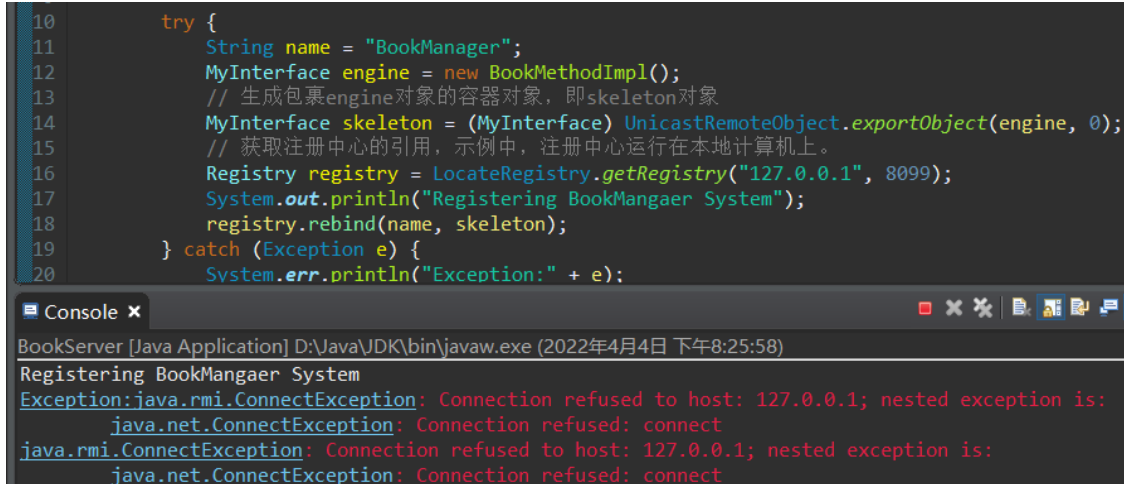
```
boolean add(Book b) // 1.添加书籍
Book queryByID(int bookID) // 2.查询书籍(通过ID)
BookList queryByName(String name) // 3.查询书籍(通过关键字)
boolean delete(int bookID) throws // 4.删除书籍
boolean setNum(int bookID,int bookNum) // 5.更新书籍库存
String showInfo() // 6.显示所有书籍
```

接口实现 BookMethodImpl 功能函数

```
// MyInterface 功能函数的基础上另有  
public void update() // 将书籍信息更新到存储信息的BookManager.txt中
```

4. 问题解决

在服务器端实现过程中，遇到了ConnectException的报错。



```
10     try {  
11         String name = "BookManager";  
12         MyInterface engine = new BookMethodImpl();  
13         // 生成包裹engine对象的容器对象，即skeleton对象  
14         MyInterface skeleton = (MyInterface) UnicastRemoteObject.exportObject(engine, 0);  
15         // 获取注册中心的引用，示例中，注册中心运行在本地计算机上。  
16         Registry registry = LocateRegistry.getRegistry("127.0.0.1", 8099);  
17         System.out.println("Registering BookMangaer System");  
18         registry.rebind(name, skeleton);  
19     } catch (Exception e) {  
20         System.err.println("Exception:" + e);  
    }
```

Console x

BookServer [Java Application] D:\Java\JDK\bin\javaw.exe (2022年4月4日 下午8:25:58)

Registering BookMangaer System

Exception:java.rmi.ConnectException: Connection refused to host: 127.0.0.1; nested exception is:
java.net.ConnectException: Connection refused: connect
java.rmi.ConnectException: Connection refused to host: 127.0.0.1; nested exception is:
java.net.ConnectException: Connection refused: connect

通过网络查询，我了解了这个问题其实是由rmi服务器端程序造成的。客户端程序向服务端请求一个对象的时候，返回的stub对象里面包含了服务器的hostname，客户端的后续操作根据这个hostname来连接服务器端。

因此我舍弃了RMI示例程序中利用IP地址和端口号获取注册中心的方法 getRegistry，使用 createRegistry，并在客户端进行了相应的更改，解决了改问题。

5. 实验测试结果

客户端测试结果截图：

Console x

BookClient [Java Application] D:\Java\JDK\bin\javaw.exe (2022年4月4日 下午10:32:25)

```
-----图书信息管理系统-----  
-----1.添加书籍-----  
-----2.查询书籍(通过ID)-----  
-----3.查询书籍(通过关键字)-----  
-----4.删除书籍-----  
-----5.更新书籍库存-----  
-----6.显示所有书籍-----  
-----else.退出系统-----  
-----
```

请输入功能选项:

6

```
-----库存书籍列表-----  
ID: 101 Name:机器学习 Author:a1 Num:3  
ID: 102 Name:分布式计算 Author:a2 Num:4  
ID: 103 Name:深度学习 Author:a3 Num:5  
ID: 104 Name:多媒体数据处理 Author:a5 Num:6
```

请输入功能选项:

1

请输入书籍信息 (书籍ID 书籍名称 作者姓名 书籍库存):

105 人工智能 a2 4

书籍添加成功!

请输入功能选项:

2

请输入需要查询的书籍ID:

101

书籍信息如下:

ID: 101 Name:机器学习 Author:a1 Num:3

请输入功能选项:

3

请输入书籍的名称关键字:

学习

```
-----查询书籍列表-----
```

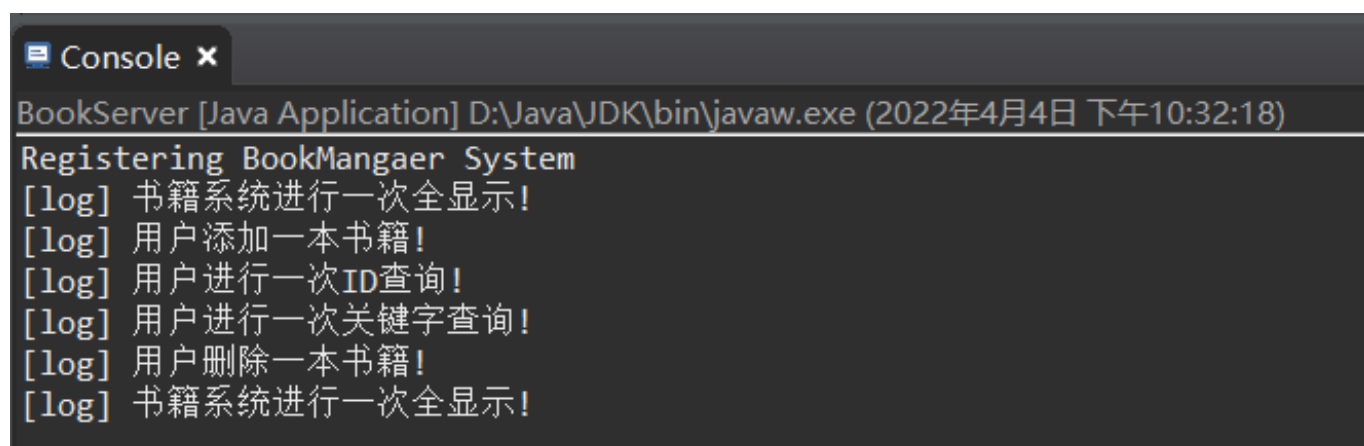
ID: 101 Name:机器学习 Author:a1 Num:3

ID: 103 Name:深度学习 Author:a3 Num:5

请输入功能选项:

```
4
请输入书籍ID:
104
书籍删除成功!
请输入功能选项:
6
-----库存书籍列表-----
ID: 101 Name:机器学习 Author:a1 Num:3
ID: 102 Name:分布式计算 Author:a2 Num:4
ID: 103 Name:深度学习 Author:a3 Num:5
ID: 4 Name:人工智能 Author:a2 Num:105
```

服务器端测试结果截图：



```
Console x
BookServer [Java Application] D:\Java\JDK\bin\javaw.exe (2022年4月4日 下午10:32:18)
Registering BookMangaer System
[log] 书籍系统进行一次全显示!
[log] 用户添加一本书籍!
[log] 用户进行一次ID查询!
[log] 用户进行一次关键字查询!
[log] 用户删除一本书籍!
[log] 书籍系统进行一次全显示!
```

从客户端的结果来看，书籍的增，删，改都完成了实现，通过ID的查询和通过关键字的查询也正确无误。可见服务器端和客户端通过RPC协作进行了信息交互，具体要求中的内容基本实现。

6. 心得体会

将上课理解的RPC机制进行了深入的亲身实践，使我对RPC的机制有了更深一层的理解，同时增强了项目能力和编程能力，受益良多。