

+

bookName

+

"\n";

第二次作业 18030100126潘银银 说明文档

#编译器&环境

notepad++ ; cmd命令行；Windows 10；

需要引入RMI 中间件

#设计说明

老师已经给出了a+b的例子 ，我们的目的是修改ComputingService接口文件 ，在里面添加 我们的方法 ，来实现需要实现的功能。

1.创建书籍对象

应题目要求 ，需要按照ID进行查询以及对Name进行魔化查找 ，所以我给书籍只增加了两个 属性 ，就是bookID 和 bookName。



private int bookID; private String bookName;

1

2

同时 ，对与每个书籍对象 ，应该有获取其ID,Name的方法以及获取该书信息的方法

1 public int getID() {

2 return bookID; 3 }

4 public String getName() {

5 return bookName; 6 }

7 public String getInfo() {

8 return "ID: " + bookID + " name:" 9 }

10 public void showInfo() {

11 System.out.println(getInfo()); 12 }

2.Java 序列化

对象序列化机制 (object serialization) 是Java语言内建的一种对象持久化方式 ，通过对象 序列化 ，可以把对象的状态保存为字节数组 ，并且可以在有需要的时候将这个字节数组通过 反序列化的方式再转换成对象。对象序列化可以很容易的在JVM中的活动对象和字节数组 (流) 之间进行转换。

老师在下课的时候提示了我们这点 ， 自然我们要在类的后面使用序列化的接口



class Book implements Serializable class BookList implements Serializable

1

2

3.存储方式

因为我是直接在notepad++中写的 ，所以为了方便起见 ，就直接将信息存在了文件中 ，非 常方便 ，连接数据库的话有点麻烦。

5.总体思路

就是在运行开始的时候先进行初始化 ，将txt文本中的书籍信息读入ArrayList列表中 ，相当 于读入内存中 ，对于每种操作 ，都会将内存中的信息写入txt文本中一直更新！

6.各种接口的实现：

boolean add(Book b) 增加一本书籍：

需要判断输入的ID是否存在 ，存在则直接返回 ，不存在则添加进入所有书的信息的

ArrayList中去。

1 for (int i = 0; i < all\_book.size(); i++) {

 2 if (all\_book.get(i).getID() == b.getID()) {

 3 return false;

 4 }

 5 }

6 all\_book.add(b);

Book queryByID(int bookID) 查询指定ID的书籍对象：

需要判断输入的ID是否存在 ，存在则将该书的消息加入book中 ，不存在返回空； 在实际的客户端通过判断返回值是否为空来打印相应的信息。

1 Book bb = null;

 2 for (int i = 0; i < all\_book.size(); i++) {

 3 if (all\_book.get(i).getID() == bookID) {

 4 bb = all\_book.get(i);

 5 return bb;

 6 }

 7 }

8 return null;

|  |  |  |  |
| --- | --- | --- | --- |
| 1  2 | //客户端判断输出提示信息  if (queryid != null) { |  |  |
| 3  4  5  6 | System.out.println(" ‐ ‐ ‐ queryid.showInfo(); } else {  System.out.println(" ‐ ‐ ‐ | ‐‐‐results ‐ ‐ ‐ ‐  ‐‐not found ‐ ‐ ‐ | ‐‐ ");  ‐‐ "); |
| 7 | } |  |  |

BookList queryByName(String name) 模糊查询书籍列表：

我采用的是字符串匹配的算法 ，根据indexOf的返回值判断 ，-1表示不匹配。

1 all\_book.get(i).getName().indexOf(name)>=0

boolean delete(int bookID) 增加一本书籍：

遍历all\_books ，存在则直接输出 ，返回正确 ，不存在则返回错误。

|  |  |
| --- | --- |
| 1  2 | for (int i = 0; i < all\_book.size(); i++) {  if (all\_book.get(i).getID() == bookID) { |
| 3  4  5  6  7 | all\_book.remove(all\_book.get(i));  savetxt();  return true;  }  } |
| 8 | return false; |

|  |  |  |
| --- | --- | --- |
| 1 | //客户端判断输出提示信息 |  |
| 2  3  4  5 | if (computingObj.delete(deleteID)) { System.out.println(" ‐‐‐‐‐delete successful ‐  } else {  System.out.println(" ‐‐‐‐‐‐delete failed ‐ ‐ ‐ ‐ | ‐‐‐ ");  ‐ "); |
| 6 | } |  |

7.一些细节：

编码问题： 由于是GBK编码 ，在我使用中文的过程中会发生错误 ，所以我删除了注释信息， 输出的提示信息全部改为了中文

运行问题：老师上课演示的是rmiregistry 8889绑定端口 ，而我在代码中直接写入 ，不用手 动命令行输入了。

#运行结果

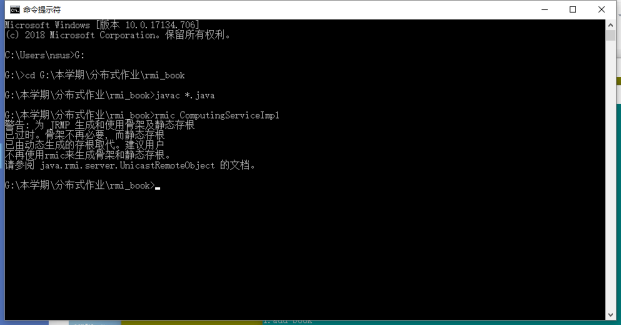
1.新建一个命令行 ，编译所有文件 ，运行中间件。

1

2

javac \*.java

rmic ComputingServiceImpl

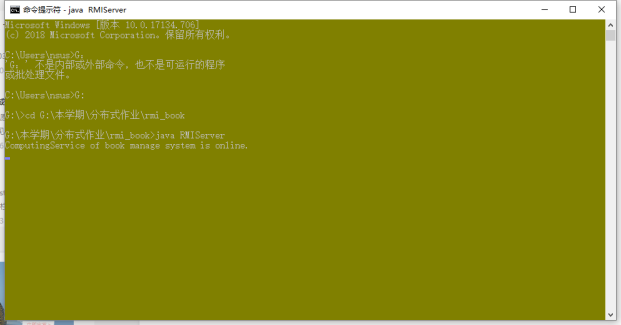


2.运行服务器端



1 java RMIServer

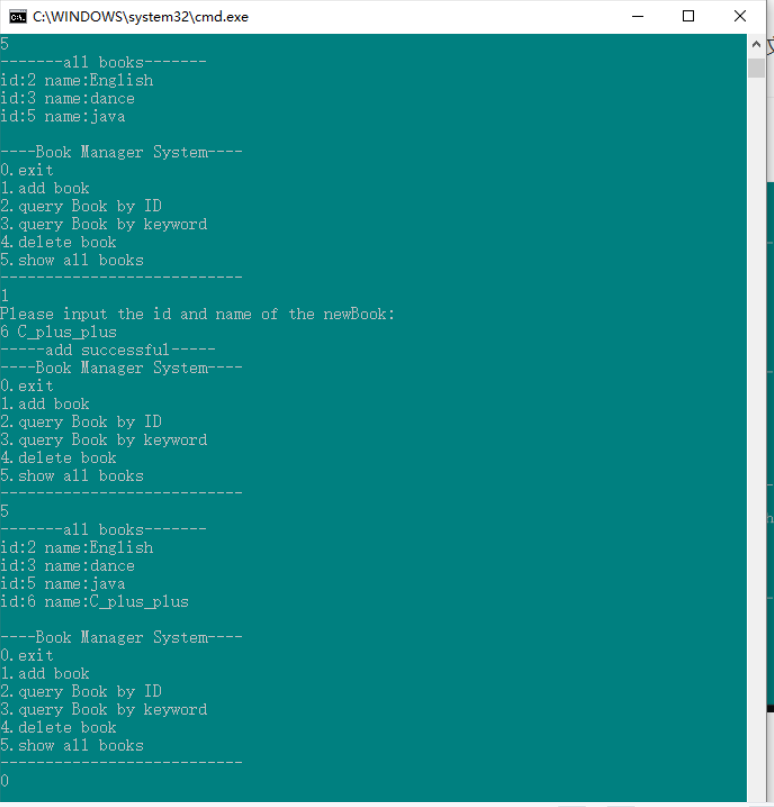




3.运行客户端

1 java RMIClient





经测试，各项功能均正确，txt文本中的书籍信息也是正常的，实验完成！