**图书管理系统**

**设计报告**

**设计思路：**

将自己提供的远程服务注册到注册中心

存储、维护和管理书籍信息系统的数据结构

查询注册中心获得远程方法的引用

调用远程方法实现与书籍管理系统的信息交互

提供用户交互界面，以处理用户的各种操作请求与输入输出。

**设计过程：**

定义书籍类Book，在其中定义书籍类属性bookID和bookName，接着定义实例 的构造方法，最后定义一个BookInfo()方法用来打印某个书籍对象的信息。

定义接口BookSystemInt，在其中声明系统需要用到的所有方法

   定义远程对象实现类BookSystemImpl，在其中实现书籍信息管理系统的数据结构 定义以及数据操纵函数

   bool add(Book b)：遍历bookList看是否存在同ID，存在则拒绝添加，返回 false；否则使用add()方法添加新书籍并返回true。

  Book queryByID(int bookID)：遍历bookList看是否存在ID=bookID的书籍，存 在则表示查询成功，返回true；否则返回false。

  BookList queryByName(String name)：遍历bookList看是否存在bookName能和 name模糊匹配成功的书籍，若存在则表示查询成功，返回true；否则返回false。  bool delete(int bookID)：遍历bookList看是否存在ID=bookID的书籍，若存在则 使用remove()方法删除指定ID的书籍并返回true；否则返回false， 表示未找到 ID=bookID的书籍，无法删除。

  bool alter(int bookID)：遍历bookList看是否存在ID=bookID的书籍，存在则用 b.bookName = newName修改指定ID书籍的书名为新的书名并返回true；否则返回 false，表示未找到ID=bookID的书籍，无法对其进行修改。

StringBuilder printBookList()：用bookList.size() == 0判断书籍列表是否为空， 若 为空返回"none"，否则遍历书籍列表，用StringBuilder.append()方法将当前书籍的信 息字符串添加到StringBuilder对象中，最后返回StringBuilder对象。

定义服务器实现类RMIServer，先定义RMI服务器IP地址、监听端口号、RMI服 务名称，再创建注册中心，使其监听来自指定端口的请求，最后将RMI服务名绑定至 远程方法实现类上。

定义客户端实现类RMIClient，在其中实现简单的用户交互界面的搭建。主要包括 显示所有的功能列表，根据用户输入的功能号调用相应的远程方法进行对书籍的增删 改查，再根据远程调用返回的结果进行后续处理等等。