10.27讨论内容总结

1. 查询函数

每次查询都要遍历所有图书，只能通过优化搜索词和单个图书名的匹配速度来提高查询效率

采用ac自动机算法 一个思路

1. 对预约问题做了详细的讨论修改

在Book类新增两个数据bookMan（记录预约该书的人数）tStorage（临时库存）

如果有人预约该书，那么被还回来的书先保存在临时库存里面，还书的时候应该先检测有没有人预约

如果有🡪检测临时库存是否等于预约人数🡪如果等于说明这时临时库存的书足够供给预约的人🡪那么直接把这一本书放在库存中，不再放入临时库存

当预约该书的人取出该书 临时库存—

当预约该书的人过了期限 update系统时把该书从临时库存转移到库存

1. 设立操作缓冲区

考虑到整个系统不能实时运作

设立一个操作缓冲区（可变更的记录）

该缓冲区只保存两种操作

格式如下：

操作类型 卡号 书号 应还日期/预约到期日期 是否已记录标识（避免重复记录）

（0表示未记录，1表示已记录）

操作类型只分两种需要更新的内容

A XXX XXX 2017-05-10 （A表示借书操作，后面是应还日期）

B XXX XXX 2017-06-10 （B表示预约操作，后面是预约失效日期）

1. 有人借书时，添加A
2. 有人预约时，添加B
3. 有人还书时，删除A
4. 有人预约到期时，写入记录，并将标识记为1
5. 用户选择删除失效预约记录，删除B
6. 用户主动取消预约，删除B

此处单独写一个用户记录检索函数 用于打印该用户目前的状态

用户登录后，检索该缓冲区

如果在缓冲区检索到该用户

操作类型为A 打印该用户借了什么书 应还日期

操作类型为B 打印该用户预约了什么书 预约状态（有效 失效 可借）

更新函数

用户登录后执行该更新函数 检索这个缓冲区

该函数主要针对日期进行操作

1. 匹配查看借书记录是否超期

如果超期 计算其违约金额 修改其违约金数额

1. 匹配查看预约记录是否失效

如果失效 写入系统记录 将该记录标识置为1

然后找到该书 令其预约人-1

如果此时临时库存>预约人数

把书放入库存 临时库存-1

1. 查找出当前用户的操作

查询用户借书状态>>

如果该用户有借书

对日期进行匹配

然后打印出来 应还日期 违约金额

查询用户预约状态>>

如果该用户有预约

匹配其日期 判断是否失效 提示该预约失效

用户选择删除预约 预约本数-1 可借本数+1

如果没有失效

查询这本书的临时库存，如果有临时库存，提示可借 跳转借书界面

如果没有临时库存 提示该预约状态有效