

优点

- 包结构清晰;
- 项目完成度很高, 整体运行下来, 如果不刻意地找bug的话, 是可以正常运行的
- 代码还是比较清晰的, 不去阅读注释也是可以读得懂的, 这是一个很好的优点
- 使用到了抽象类和继承, 抽象类的功能实现的还是比较ok的
- 使用到了抽象工厂设计模式, 这一点是很不错的, 通过实现一个factory接口来完成不同工程的分工化创建

优点总结: 师弟或师妹在这一个项目付出的精力和时间是可以看得到的, 项目当中所体现的java基础语法是比较扎实的, 同时, 师弟或师妹在一些输入值的格式匹配和输入值的提醒词语是做的比较好; 而且, 一整个订房业务下来, 也不会出现一些致命的异常, 说明师弟或师妹在设计和运行的时候一定是有大量的debug调式的, 这一点做得很好; 师弟或师妹继续加油!!!

缺点

- 注册环节有一个小缺点: 用户名输入的时候没有说明不能输入空格, 说明当用户输入空格的时候并且在五个字符之内, 是可以通过的; 这样子是不行的
 - 解决方案: 使用正则表达式或设计一种算法来识别空格字符串, 确保用户名的格式不会出现空格字符
- 注册成功之后, 用户需要记住其系统分配的ID值, 这样子对用户不是很友好; 一般在用户名设计环节, 登录环节都是采用的是账户名登录的方式, 而ID这种一般都是用来进行业务操作的; 不会一开始就使用在用户登录上
- 而且, 在登录环节, 账户登录提示说只能输入3位数的整数, 那么假如该程序注册的账号已经是大于999的, 那该怎么办
- 开房成功之后, 一旦涉及到的减法操作出现小数 就会造成这样子的意外
 - 欢迎用户: amour 当前余额: 199.90000000000001 ID: 1
 - 解决方案: 可以采用的是BigDecimal这个类来进行格式小数处理, 确保最终展示的小数点只有两位;
- 开房环节有一个小缺陷: 用户没有办法开多个房间, 这其实在现实是行不通的, 而且在网上的程序, 订房是可以同时订多间的, 这个需要注意一下

缺点总结: 大部分都是业务逻辑和格式问题;

