一评

优点

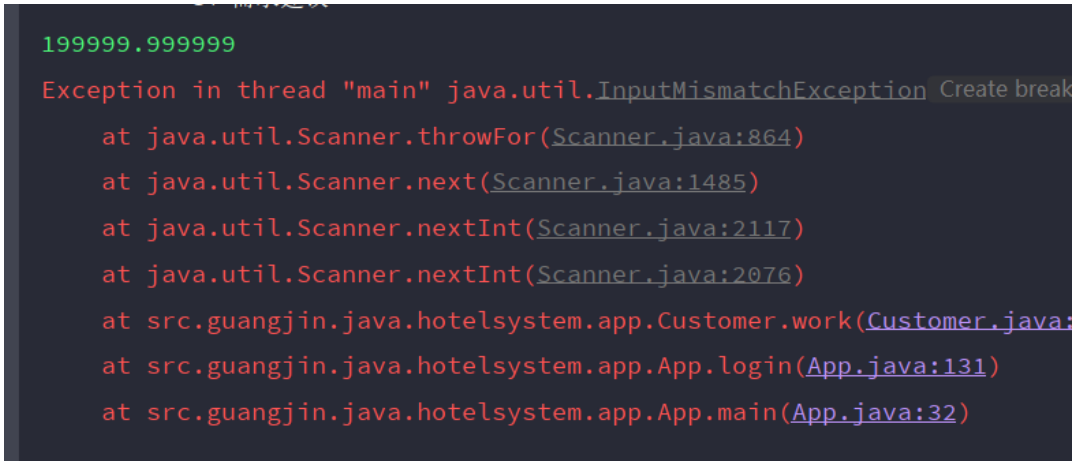
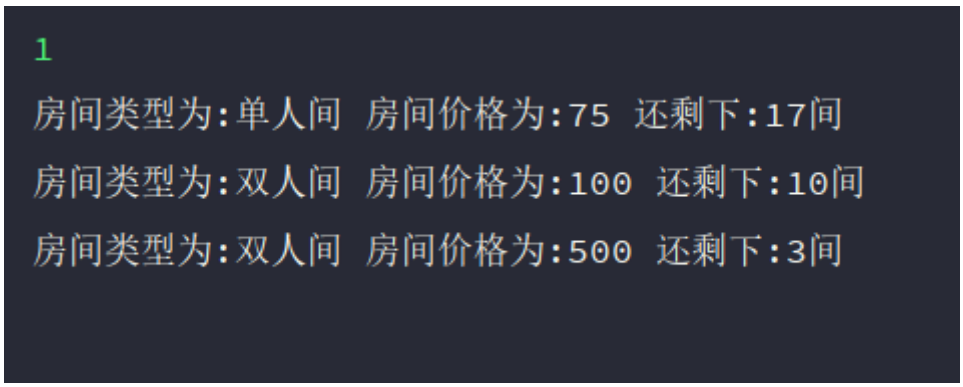
- 项目完成度很高，代码可读性很强，在注释方面，做到了短小而又精悍；看出师弟或师妹在这一个考核任务上面是十分用心的；
- 在类设计上，类名见名知义；每一个类都各司其职，做到了功能模块化设计；
- 采用了抽象类的思想；在程序设计环节，也很好地将抽象类的一些功能做到很好；
- 程序运行环节给出的提示还是比较可以的，可以让用户知道这一个选项所需要使用的业务；

优点总结

- Java语法了解的很好，同时也有将一些自己的思想同Java语法相结合，说明师弟或师妹在这一个基础的学习是比较扎实，对于师弟这一阶段的学习，我觉得师弟做得比我好多了，毕竟在师弟这一个阶段我可能还不会使用到抽象里来进行类设计；
- 师弟的整一个业务逻辑是十分清晰的，同时代码在运行环节可以让我们这些人一下子就找到了地方，这一方面，师弟在程序设计方面也是不错的；
- 有一个小小的建议：虽说师弟现在用到的类是比较少的，但是建议师弟根据每一个类的功能来进行包分类；

缺点

- 注册登录环节少了一个用户提示的细节，在用户注册的用户名设置和密码设置当中没有提醒用户名的规范和密码的规范；
    - 解决方案：可以了解一下正则表达式和输入字符串之间的匹配规则，建议在注册环节用户名和密码的格式上做一点适当的提示
    - 风险：当用户不了解一个程序的用户名和密码格式的规则，那么易造成程序bug出现，假如加上了提示和匹配规则，那么在一定程序上可以减少风险的出现，提升了容错率；
- 充值环节下有一个小缺点，同样是没有提示，没有明确规定用户可以充值的金额是整数还是可以带小数点的小数

- 
```
199999.999999
Exception in thread "main" java.util.InputMismatchException Create break
    at java.util.Scanner.throwFor(Scanner.java:864)
    at java.util.Scanner.next(Scanner.java:1485)
    at java.util.Scanner.nextInt(Scanner.java:2117)
    at java.util.Scanner.nextInt(Scanner.java:2076)
    at src.guangjin.java.hotelsystem.app.Customer.work(Customer.java:
    at src.guangjin.java.hotelsystem.app.App.login(App.java:131)
    at src.guangjin.java.hotelsystem.app.App.main(App.java:32)
```
  - 风险：造成运行时异常的出现，降低程序运行的效率
  - 解决方案：在充值环节可以采取设置一定额度的充值金额，用户自行选择哪一个选项，减少了用户自己输入金额时造成的bug缺陷
- 开房业务有一个小缺点：当用户多次订购多个房间的时候，是什么东西可以区分用户订购的房间之间的区别，没有一个标识来区分房间的信息 例：房号等；在选择查看自己的房卡的时候 显示的只是订购了一个单人间而已；
  - 解决方案：通过在room这一个实体类下设置一个具有唯一性的唯一性属性（建议：房间号）；
- 退房业务有一个大漏洞：**无中生有**
  - 
```
1
房间类型为:单人间  房间价格为:75  还剩下:17间
房间类型为:双人间  房间价格为:100  还剩下:10间
房间类型为:双人间  房间价格为:500  还剩下:3间
```
  - 风险：造成线上开房业务的房间信息和线下的房间信息不匹配；
- 切换账户业务有一个缺点：在切换账户之后，发现里面的金额仍然是共享的，那么问题就是：你是怎么区分这两个账户是同一个主人的呢；而且，当我注册另外一个用户名的时候，发现在登录这一个新注册的用户名的时候仍然是之前的一个用户；

```
请输入用户名
username1
请输入密码
1234561
登录成功，可以开始使用小帮手了



######欢迎使用旅馆小助手,亲爱的顾客#####
欢迎会员:username      当前余额:119805    享受折扣:八折
----------1、所有房型的信息----------
----------2、开房----------
----------3、退房----------
----------4、查看自己的房卡----------
----------5、充值----------
----------6、办卡----------
----------7、切换用户----------
----------8、需求建议----------
```

缺点总结：大部分都是业务的逻辑出现了问题，在程序的代码方面，发现在一些设计的时候都是可以的，但可能是因为缺少开发酒店订房系统的经验，导致一部分业务逻辑在现实环境下问题较大