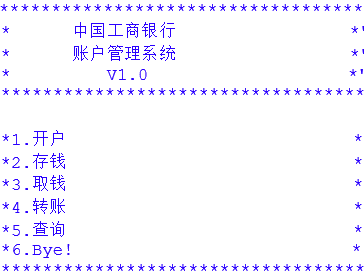
面向对象

1. 仔细分析需求完成下列功能实现：
   1. **用户类**：账号（int）、姓名(Str)、密码(8位数字)、地址、存款余额(double)、注册时间(date)、开户行（银行的名称（String））
   2. **地址类**：国家(String)、省份(String)、街道(String)、门牌号(String)
   3. **银行类**：能存储100个用户的库([100])、本银行名称（比如：中国工商银行的昌平支行,String）
      1. 银行业务功能
         1. 添加用户（传入参数：**用户**。返回值：**整型值**（1：成功，2：用户已存在，3：用户库已满））
            1. 业务逻辑：

* 先检查该用户的账号在库里是否存在。若不存在则在用户库里添加一个该用户并返回代号1，
* 若存在则返回代号2。另外在添加用户的时候检测用户库是否已注册满，若已满则返回代号3
  + - 1. 存钱（传入值：用户的账号。存取的金额。返回值：布尔类型值）
         1. 业务逻辑：
* 先根据传入的账号信息查询用户库里是否有该用户。若没有则返回false
* 若有，则将该用户的金额存进去。
  + - 1. 取钱（传入值：用户的账号，用户密码，取钱金额。返回值：整型值（0：正常，1：账号不存在，2：密码不对，3：钱不够））
         1. 业务逻辑：
* 先根据账号信息来查询该用户是否存在，若不存在，则返回代号1，
* 若存在，则继续判断密码是否正确，若不正确，则返回代号2。
* 若账号密码都正确，则继续判断当前用户的金额是否满足要取出的钱，若不满足，则返回代号3，
* 若满足，则将该用户的金额减去。
  + - 1. 转账（传入值：转出的账号，转入的账号，转出账号的密码，转出的金额。返回值：0：正常，1：账号不对，2密码不对，3钱不够）
         1. 业务逻辑：
* 先查询用户库是否存在转出和转入的账号，若不存在则返回代号,1，
* 若账号都存在则继续判断转出账号的密码是否正确，若不正确，则返回2，
* 若正确则继续判断要转出的金额是否足够，若不够则返回3，
* 否则正常转出，转出的账号用户金额要相对应的减少，转入的金额相对应的增加。
  + - 1. 查询账户功能（传入值：账号，账号密码，返回值：空）
         1. 业务逻辑：
* 先根据账号判断用户库是否存在该用户，若不存在则打印提示信息：该用户不存在。
* 否则继续判断密码是否正确。若不正确则打印相对应的错误信息。
* 若账号和密码都正确，则将该用户的信息都打印出来，比如：当前账号：xxxx,密码:xxxxxx,余额：xxxx元，用户居住地址：xxxxxxxxxxxxx，当前账户的开户行：xxxxxxxxxx.
  1. **界面类**：在执行该入口程序时，就打印银行业务选择菜单：比如：
     1. 
     2. 然后就开始处理各种输入操作，直到业务处理完成!

答案：