**项 目 报 告**

指导老师: 张 文 广

姓 名: 康 竞 豪

班 级: python21

目 录

[一、 main模块：系统主界面模块 3](#_Toc11268465)

[二、 view模块：登录界面模块 3](#_Toc11268466)

[三、 card模块：存储卡的信息模块 3](#_Toc11268467)

[四、 person模块：存储用户信息模块 3](#_Toc11268468)

[五、 operation模块：具体功能实现模块 3](#_Toc11268469)

[六、 各个方法实现时参数的逻辑判断，检测的方法： 6](#_Toc11268470)

[七、 项目总结 7](#_Toc11268471)

项目功能汇总:

本项目是基于python基础语言写的是一个与银行相关的各种操作功能，具体功能有：注册，查询，存款，取款，转账，更换密码，锁卡，解卡，补卡，退出保存。

1. **main模块：系统主界面模块**

功能：所有的函数，实现方法都在这个模块里调用实现，通过导入包的单入口模式调用各个模块的成员方法、属性以实现相应的功能

1. **view模块：登录界面模块**

功能：登录界面判定默认为，管理员账户：admin，密码123

实现：生成一个class View类，创建一个run（）方法，run的功能是判断默认管理员的账户：admin，密码123

1. **card模块：存储卡的信息模块**

定义一个Card类，初始化变量：卡号， 密码， 余额， 是否锁定

cardid password money islock

1. **person模块：存储用户信息模块**

定义一个Person类，初始化变量：用户名，身份证号，手机号， 卡

Name userid phone card

1. **operation模块：具体功能实现模块**
2. register方法：注册功能方法实现

注册功能，输入用户名、身份证号、手机号、判断是否合法，调用get\_pwd方法得到密码，银行卡余额为10，实例化银行卡类和用户类，添加到字典。我们所要知道的必要参数有：用户名，身份证号，手机号，密码这四项基本数据。其中限定条件：

用户名name：不能为空，且不能含有空空格

身份证号userid：18位，且为纯数字类型

手机号phone：11位，且为纯数字类型

初始余额money：指定新开用户余额为10

密码password：指定为6位，纯数值类型，并且重复确认的密码要和第一次相同

银行卡号cardid：生成的银行卡账号，必须是随机6位，且不重复

1. query方法：查询功能方法实现

查询功能：输入想要查询的卡号,调用get\_card\_info判断银行卡号是否存在，存在判断islock的状态，没锁定则得到卡内余额，这里需要用到两个参数：银行卡号和余额。其限定条件有：

银行卡号：必须是存在的银行卡号

余额：指定新开用户的余额为10，且余额必须大于10

1. save\_money方法：存款功能实现方法

存款功能：输入要存入的账号，调用self.get\_card\_info判断银行卡号是否存在，存在判断islock的状态，没锁定则输入存入金额，并打印结果，这里有三个参数：银行卡号，存入的金额，原本剩下的余额，其限定条件有：  
银行卡号：必须是已存在的银行卡号

存入的金额：输入存入的数值

原本剩下的余额：调用原来的参数

打印出来的现有金额 = 存入 + 余额

1. get\_money方法：取款功能实现方法

取款功能：输入要取款的账号，调用self.get\_card\_info判断银行卡号是否存在，存在判断islock的状态，没锁定则输入取款金额，然后判断余额减输入值大于10，打印结果，这里有三个参数：银行卡号，取出的金额，原本剩下的余额，其限定条件有：

银行卡号：必须是已存在的银行卡号

取出的金额：输入存入的数值

原本剩下的余额：调用原来的参数

打印出来的现有金额 = 余额 – 取出 ，此时的结果要大于10

1. trans\_money方法：转账功能实现方法

转账功能：输入自己的银行卡，调用self.get\_card\_info判断银行卡号是否存在，存在判断islock的状态，没锁定则输入密码调用check\_pwd判断密码，输入对方的银行卡号，调用self.get\_card\_info判断银行卡号是否存在，存在判断islock的状态，没锁定则输入转账金额，打印双方余额（现实中不会显示对方的余额，这里为了方便查看整个转账过程，选择显示对方余额），这里有个参数：银行卡1，银行卡1密码，银行卡2，转账金额，银行卡1余额，银行卡2余额，其限定条件有：

银行卡1：必须是已存在账号

银行卡1密码：必须与银行卡1密码一致

银行卡2：必须是已存在账号 （判断银行卡1是否等于银行卡2）

转账金额：要转账的金额

银行卡1余额：银行卡1原余额 - 转账金额

银行卡2余额：银行卡2原余额 + 转账金额

1. change\_money方法：更换密码实现方法

改密功能：输入卡号，调用self.get\_card\_info判断银行卡号是否存在，存在判断islock的状态，没锁定则选择用身份证或原密码改密，输入新密码调用get\_pwd方法判断是否符合密码规范，用新密码覆盖原密码的整个过程。这里的参数有：银行卡号，原密码，身份证号，新密码，其限定条件有：

银行卡号：必须已存在

原密码：必须与银行卡号匹配

身份证号：必须与银行卡号匹配

新密码：必须为6位纯数值

确认密码：必须与新密码一致

1. locck方法：锁卡实现方法

冻结功能：输入卡号，调用self.get\_card\_info判断银行卡号是否存在，存在判断islock的状态，没锁定则选择原密码/身份证号锁卡，若之前就是锁定状态，则不操作。这里的参数有：

银行卡号：必须已存在

原密码：必须与银行卡号匹配

身份证号：必须与银行卡号匹配

1. unlock方法：解卡实现方法

解卡功能：输入卡号，调用self.get\_card\_info判断银行卡号是否存在，存在判断islock的状态，没锁定则选择原密码/身份证号解卡，若之前就没有的话，则不操作。这里的参数有：

银行卡号：必须已存在

原密码：必须与银行卡号匹配

身份证号：必须与银行卡号匹配

1. new\_card方法：补卡实现方法

补卡功能：输入身份证号，调用get\_userid方法判断身份证号是否合法，合法则进行补卡，调用get\_cardid方法生成新的银行卡号，这里的参数有：

身份证号：必须是已存在的

补办的卡号：随机生成，且与之前的不重复

1. save方法：退出保存实现方法

保存退出功能：利用文件操作，打开相关文档，若不存在则创建文件，利用pickle模块的load/dump对文件进行读取写入保存操作

1. **各个方法实现时参数的逻辑判断，检测的方法：**
2. get\_name方法：判断名字是否合法

对名字name进行if条件判断：如果名字为空“” 或者 name 中含有空格”“，打印”输入名字不合法“，否则将name返回出来。

1. get\_userid方法：判断身份证号是否合法:

首先判断该身份证有没有注册过，如果有，则提醒：“已注册，忘记可补卡”

然后进行长度len（）判断：如果输入不是18位，报错；是18位，检测是否有“”空格，有则报错，没有判断是否为纯数值，不是报错，是将身份证号返回出来。

1. get\_phone方法：判断手机号长度  
   首先进行长度len（）判断：如果输入不是11位，报错；是11位，检测是否有“”空格，有则报错，没有判断是否为纯数值，不是报错，是将身份证号返回出来。
2. get\_pwd方法：判断密码是否相等  
   首先进行长度len（）判断：如果输入不是6位，报错；是6位，检测是否有“”空格，有则报错，没有判断是否为纯数值，不是报错，是纯数值则判断第二次和第一次是否相等，相等将身份证号返回出来，否则打印错误。
3. get\_cardid方法：获取卡号  
   用random.randint（）模块方法生成一个随机卡号，然后检查这个卡号是否是已存在的，存在则报错，不存在则返回生成的这个卡号。
4. check\_pwd方法：检测密码，不对的话就锁卡写一个3次循环，判断密码与卡号是否匹配，不匹配的话，让islock=True，进行锁卡。
5. get\_card\_info方法：检测卡是否存在

输入一个卡号，检测卡号是否存在：

if card\_id in self.user\_dict:

1. load\_user方法：读取用户文档

打开一个user.txt的文档，先用exist进行判断，存在则打开将user\_dict字典存进去，不存在则创建并输入一个空字典

1. load\_user\_id方法：读取用户id文档

打开一个userid.txt的文档，先用exist进行判断，存在则打开将user\_id\_dict字典存进去，不存在则创建并输入一个空字典

## 项目总结

通过这次ATM系统实现项目，让我对python基础语法有了更深刻理解，在这次小项目中，基础语法的循环与判断贯穿整个项目，让我更加熟练的掌握了while循环，if-else条件判断的相关知识，对这些基础语法有了进一步的了解与认识。

在ATM操作系统中主要用到的是面向对象相关的知识，创建不同的类，对其中的类/对象和成员属性、成员方法进行调用，反复使用的过程中，让之前一知半解的知识变得更加的清晰，容易理解，当然还有很多地方需要自己再去理解和记忆。

以及单入口模式的导入模块，与使用导入模块里面的成员方法，在这个过程中，理解了单入口模式的作用，以及各个模块之间的联系，即导入相关模块，我理解为调用，比刚刚接触的时候知道了更多的相关使用方法。

还有文件操作的部分内容，利用pickle内置模块的序列化与反序列化对数据进行读取与存储，复习了之前学习的内容，让整个知识链连接了起来，学有所用，将学到的知识实例化之后，印象更加深刻，而且更容易理解。

在整个开发过程中，也遇到一些问题，变量和代码量变多的时候，容易造成思维混乱，有时候会使用错变量，导致整个项目的节奏变慢，敲代码还是需要细心，严谨。

最后还是要感谢我的指导老师张文广，通过这一个月的教导，让我对python 这门语言有了一个新的认识，“万物皆对象”，从听到敲代码到理解，这个过程中有不理解的地方，都能得到及时的指导，没有积累不会的内容拖到以后再理解。