






Python04模拟ATM电子银行

Python

一,程序文件说明

 bin	2016/7/10 14:09	文件夹
 crontab	2016/7/10 14:09	文件夹
 db	2016/11/15 4:26	文件夹
 lib	2016/7/10 12:07	文件夹
 src	2016/7/10 11:52	文件夹

程序分为5个组成部分

- bin：放置Python程序的启动接口文件

通过Python命令启动文件夹内文件即正常执行Python程序
例如：ATM_start.py

- crontab：放置Python程序的计划任务启动接口文件

通过配置计划任务定时启动Python脚本的程序接口文件
例如：Auto_mission.py

- db：放置Python程序所需的配置文件

程序生成的用户配置文件
例如：姓名+ '_log'

- lib：放置Python程序所需的公共方法文件

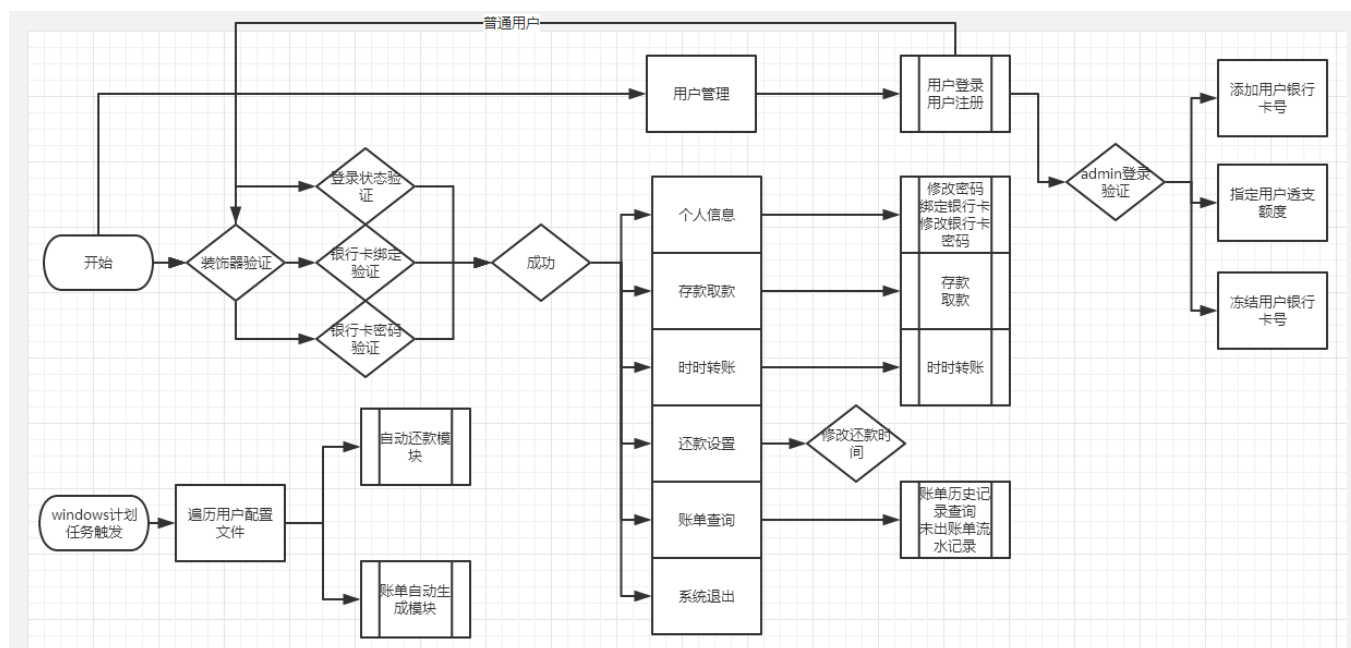
Python代码的公共方法层文件
例如：common.py

- src：放置Python程序所需的业务层文件

Python代码的业务层文件

例如：admin_business.py user_business.py

二，程序流程简图



三，用户配置文件详解

```
dict_info = {'name_login':name_login, #用户名
             'pwd_login':pwd_login,   #登录密码
             'account':account,        #银行卡号
             'pwd_money':pwd_money,    #银行卡密码
             'tm_text':tm_text,         #注册时间
             'billing_day':'15',        #账单日
             'Repayment_date':'1',     #还款日
             'status':'0',             #账户状态（0：活跃 1：冻结）
             'cash':'0',               #现金电子余额
             'Actual_overdraft':'0',   #总透支金额
             'Overdraft_limit':'15000', #透支额度上限
             'Debt_Bill_amount':'0',   #账单欠账金额记录
             'Debt_record':{},         #已出账单历史记录
             'Has_been_out_billing':{}, #已出账单流水历史记录
             'Not_out_billing':[]      #未出账单流水记录
            }
```

1.name_login:用于登录的用户名

2.pwd_login:用户登录时的密码

- 3.account:用户银行卡号
- 4.pwd_money:用户银行卡密码
- 5.tm_text:用户注册时的时间
- 6.billing_day:生成账单时间
- 7.Repayment_date:最后还款时间
- 8.status:银行账户状态
- 9.cash : 电子现金余额
- 10.Actual_overdraft:总透支金额
- 11.Overdraft_limit:透支额度上限
- 12.Debt_Bill_amount:账单欠账金额纪录（累加）
- 13.Debt_record:已出账单历史记录（里面仍旧是个字典，可以按照账单时间查询。例如：
2016-11 : str）
- 14.Has_been_out_billing:已出账单流水历史记录（里面仍旧是个字典，可以按照账单时间查询，每次生成新账单时，将未出账单流水记录里的列表内容拷贝过来并清空未出账单流水记录列表里的内容）
- 15.Not_out_billing:未出账单流水记录（用户的所有操作记录在内）

四，程序测试样图

```
E:\python2.7.8\python.exe E:/pyhomework/第二关/python04/bin/ATM_start.py

          欢迎光临ATM电子银行          用户状态未登录...

          1，用户管理
          2，个人信息
          3，存款取款
          4，时时转账
          5，还款设置
          6，查询账单
          7，退出系统

请输入索引进行选择：|
```

- 1, 用户管理
- 2, 个人信息
- 3, 存款取款
- 4, 时时转账
- 5, 还款设置
- 6, 查询账单
- 7, 退出系统

请输入索引进行选择: 6

账单功能如下: chensiqi登录中...

- 1, 账单查询
- 2, 未出账单流水记录查询
- 3, 返回菜单

请输入索引进行选择: 1

- 1, 2016-7账单
- 2, 2016-10账单
- 3, 2016-11账单
- 4, 2016-8账单
- 5, 2016-9账单

请输入你的选择: 3

亲爱的chensiqi, 您的2016年11月账单如下:

账单总金额为:11000(当期+历史欠账), 最后还款日为下月1日
请您按时还款, 谢谢您的使用, 再见!

2016-11月账单流水信息如下:

2016-11-5 3:39: 您进行了“存款”操作, 存款金额为8000, 电子现金余额为8000, 总透支额度为8000
2016-11-5 3:39: 您进行了“提款”操作, 提款金额为1000, 现金余额为7000, 总透支金额为8000
2016-11-5 3:39: 您进行了“提款”操作, 提款金额为10000, 电子现金余额为0, 总透支金额为11000
2016-11-5 3:39: 您进行了“存款”操作, 存款金额为6000, 电子现金余额为6000, 总透支额度为11000

请输入索引进行选择:

- 2, 2016-11账单
- 3, 2016-10账单
- 4, 2016-8账单
- 5, 2016-9账单

请输入你的选择: 2

亲爱的chensiqi, 您的2016年10月账单如下:

账单总金额为:3000(当期+历史欠账), 最后还款日为下月1日
请您按时还款, 谢谢您的使用, 再见!

2016-10月账单流水信息如下:

2016-11-15 4:25: 您进行了“存款”操作, 存款金额为8000, 电子现金余额为14000, 总透支额度为11000
2016-11-15 4:28: 银行卡为1111111111111111的用户 向您转账4000元, 电子现金账户余额为18000, 总透支额度为11000
2016-11-1 10:37: 您进行了“提款”操作, 提款金额为10000, 电子现金余额为0, 总透支金额为3000
2016-11-1 10:37: 您进行了“存款”操作, 存款金额为5000, 电子现金余额为5000, 总透支额度为3000
2016-10-15 10:43: 您进行了“转账”操作, 转账金额为1000, 对方银行卡号为1111111111111111, 电子现金余额为4000, 总透支额度为3000

请输入索引进行选择:

- 5, 还款设置
- 6, 查询账单
- 7, 退出系统

请输入索引进行选择: 2

您的个人注册信息如下:

登录名: chensiqi
银行卡号: 6226000077778888
注册时间: 2016-11-5 2:55
账单日(每月): 15
还款日(每月): 10
银行卡状态: 正常
电子现金余额: 1000
银行卡已透支额度: 0
银行卡透支额度上限: 15000

您可以进行如下操作:

- 1, 修改登录密码
- 2, 绑定银行卡
- 3, 修改银行卡密码
- 4, 返回菜单

你想做什么?

4: Run 6: TODO Terminal Python Console
Run AIM_start (1)

4, 系统退出

管理员你好, 你想做什么?: 3

所有已关联银行卡的用户的银行卡状态信息如下:

- 1 登录名称: chensiqi1 账户状态: 活跃
- 2 登录名称: chensiqi2 账户状态: 活跃
- 3 登录名称: chensiqi 账户状态: 活跃

输入索引选择用户来改变他的银行卡状态(活跃->冻结->活跃)(n = 返回上级菜单):

4: Run 6: TODO Terminal Python Console

管理员你好, 你想做什么? : 2

所有用户额度信息如下:

1 登录名称: chensiqi1 透支额度为: 15000

2 登录名称: chensiqi2 透支额度为: 15000

3 登录名称: chensiqi 透支额度为: 35000

输入索引选择你想修改额度的账户(n = 返回上级菜单):