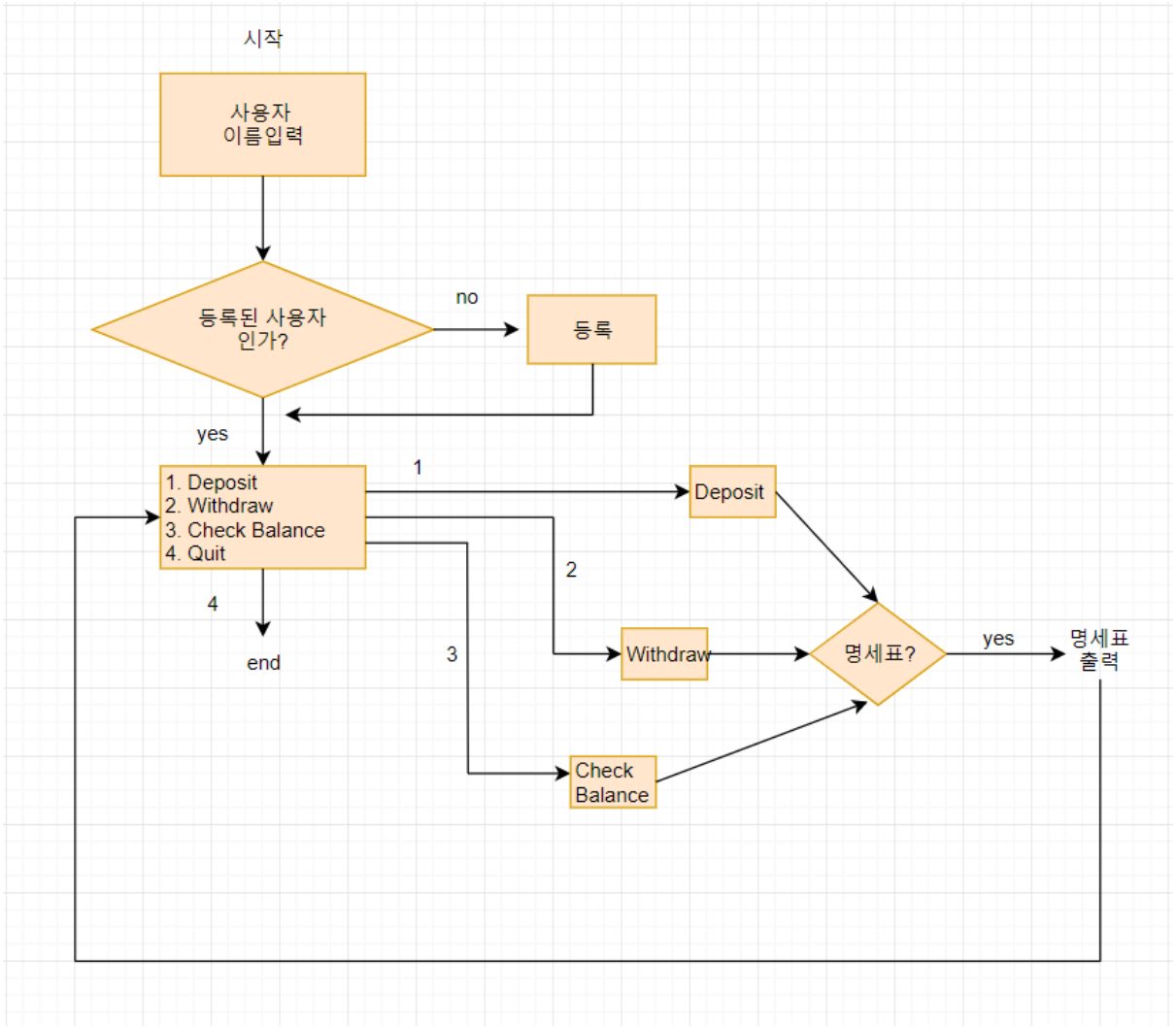
프로그래밍 입문 – 기말 프로젝트

1616557 IT공학과 박아정

1. Flowchart



2. 코드 설명

A. classBankAccount.py

- 입금/출금 기능이 있는 은행 계좌 클래스 BankAccount가 있는 파일이다.

B. classATM.py

- 은행 계좌 클래스인 BankAccount를 상속받는 클래스인 Atm이 정의되어 있는 파일이다.
- classATM.py에는 다음과 같은 기능이 구현되어 있다.

- 엑셀파일 "UserList.xlsx"로부터 고객 정보 리스트를 받아온다.
- 클래스 Atm은 사용자 이름을 입력 받은 뒤 등록된 사용자인지 아닌지 검사한다.
- 등록된 사용자일 경우 비밀번호를 입력 받고 거래를 시작한다.
- 1~3거래를 한 뒤 명세표를 출력하고 싶다면 출력해준다.
- 4를 눌러 거래를 마친다.
- 만약 등록되지 않은 사용자일 경우 비밀번호와 잔액을 입력 받은 뒤 엑셀에 새로운 사용자를 등록한다
- 이어서 똑같이 1~3에서 거래를 하고 명세표를 출력해준다.
- 4를 눌러 거래를 마친다.

- 변수 설명

filename	엑셀 파일을 받아올 변수
max_row, max_col	엑셀의 행과 열의 개수
name_list[], pwd_list[], balance_list[]	이름, 비번, 초기금액이 들어갈 리스트
name	입력한 이름이 들어갈 변수
pwd	입력한 비밀번호가 들어갈 변수
index	현재 이름이 name_list에 몇 번째 인덱스인지 알려주는 변수
balance	현재 잔액을 알려주는 변수
num	거래(1,2,3,4)중 선택을 위한 변수
amount	출금/입금할 금액을 입력받는 변수
time	현재 시간을 받아오는 변수
receipt	명세표를 출력할지 여부를 받아오는 변수

- 클래스 Atm 함수설명

__init__(self, name = 'None', pwd = 0, balance = 0)	사용자가 이름을 입력하면 등록된 사용자인지 확인, 비밀번호를 입력하고 거래를 진행한다. 만약 새로운 사용자일 경우 등록을 한 뒤 거래를 진행한다.
menu(self)	메뉴를 출력해준다.
deal(self)	선택한 메뉴에 따라서 거래를 진행한다.
get_recipt1(self, amount, balance)	입금할 때 명세표 출력
get_receipt2(self, amount, balance)	출금할 때 명세표 출력
reg_customer(self)	새로운 사용자를 등록

- 코드 설명

```
import openpyxl
import datetime
from classBankAccount import *

time = datetime.datetime.now()      #현재시간 받아옴

#엑셀파일 읽어옴
filename = "UserList.xlsx"
wb_obj = openpyxl.load_workbook(filename)
sheet_obj = wb_obj.active

max_row = 7  # 행의 개수
max_col = 3  # 열의 개수
currentCell_obj = sheet_obj.cell(row=1, column=1)
name_list = []  #이름이 들어갈 리스트
pwd_list = []    #비번이 들어갈 리스트
balance_list = [] #초기금액 들어갈 리스트

for i in range(2, max_row + 1):  # 전체 행에서
        for j in range(1, 2):  # 이름까지만
            currentCell_obj = sheet_obj.cell(row=i, column=j)
            name_list.append(currentCell_obj.value)  # 셀의 value 를
Score_list 에 추가한다.
        for k in range(2,3): # 비밀번호 저장
            currentCell_obj = sheet_obj.cell(row=i, column=k)
            pwd_list.append(currentCell_obj.value)
        for m in range(3,4): # 초기 잔액 저장
            currentCell_obj = sheet_obj.cell(row=i, column=m)
            balance_list.append(currentCell_obj.value)

class Atm(BankAccount):
    def __init__(self, name = "None", pwd = 0, balance = 0):
        self.name = input("Enter the username: ")        #이름 입력
        if self.name in name_list:      #등록된 사용자일 경우
            print(self.name, '님 환영합니다.')
            pwd = int(input(self.name + "님의 비밀번호를 입력하세요:
"))
            self.pwd = pwd
            index = name_list.index(self.name)  #입력받은 이름이
name_list 의 몇번째 인덱스인지 반환
            self.balance = balance_list[index]  #입력받은 이름에
해당되는 balance 를 가져옴
            if (self.pwd == pwd_list[index]):   #pwd_list 의
```

```
비밀번호와 비교
                print("사용자 정보가 확인되었습니다.")
                Atm.deal(self)      #거래 시작
            else: # 비밀번호가 틀렸을 경우
                print("비밀번호가 틀렸습니다.")
        else:       #새로운 사용자일 경우
            register = input(self.name + "님은 등록되지 않았습니다.
추가하시겠습니까?")
            if(register == "yes"):
                Atm.reg_customer(self)  #회원 등록
                BankAccount.balance = self.balance #입력받은 잔액
저장
                Atm.deal(self)       #거래 시작
            else: # 회원 등록 하지 않을 때
                print("거래를 종료합니다.")

    def menu(self): #메뉴를 출력해주는 함수
        print("=" * 25)
        print("원하시는 메뉴를 선택하세요.")
        print("1. Deposit")
        print("2. Withdraw")
        print("3. Check Balance")
        print("4. Quit")
    def deal(self): # 거래를 위한 함수
        while(1):   #4를 입력하기 전까지 무한 루프
            Atm.menu(self) #메뉴를 출력
            num = input(">>")
            if(num=="1"):
                amount = int(input("입금하실 금액을 입력하세요: "))
                BankAccount.balance = self.balance
                BankAccount.deposit(BankAccount, amount)
#부모클래스의 deposit 함수를 불러온다.
                receipt = input("명세표를 출력하시겠습니까? ")
                if(receipt=="yes"):     #명세표 출력하고 싶다면
                    Atm.get_receipt1(self, amount,
BankAccount.balance)     #명세표 출력
                else:
                    print("이어서 거래를 진행하세요.")
                self.balance = BankAccount.balance #처리한 결과를
저장함
            elif(num=="2"):
                amount = int(input("출금하실 금액을 입력하세요: "))
                BankAccount.balance = self.balance
                print(BankAccount.balance)
                BankAccount.withdraw(BankAccount, amount)
```

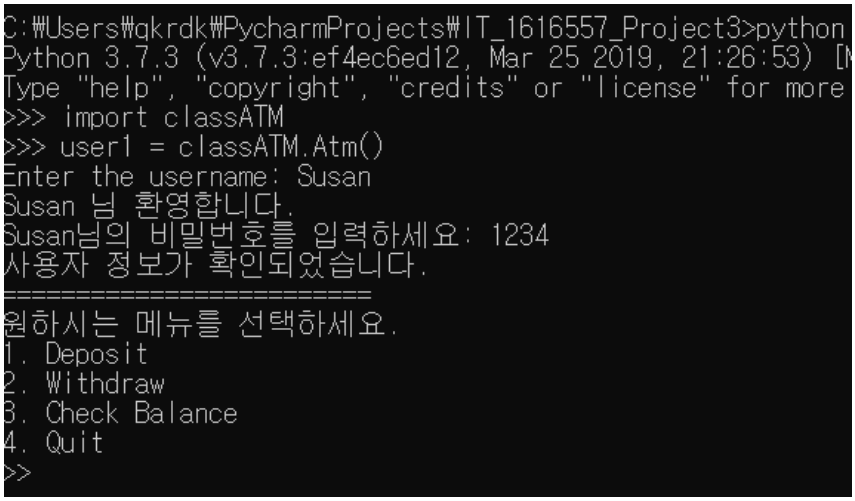
```
#부모클래스의 withdraw 함수를 불러온다.
                receipt = input("명세표를 출력하시겠습니까? ")
                if (receipt == "yes"):
                    Atm.get_receipt2(self, amount,
BankAccount.balance)
                else:
                    continue
            elif(num == "3"):
                print("현재 잔액은", BankAccount.balance, "입니다.")
#앞에서 저장한 현재 잔액을 가져옴
            elif(num == "4"):
                break  #while문 빠져나가기

    def get_receipt1(self, amount, balance): #입금을 위한 명세표
        print("*"*25)
        print("{0:=^26}".format("명세표"))
        print("거래시간: ", time)
        print("이름: ", self.name)
        print("입금액: ", amount)  #입금한 amount를 가져옴
        print("남은 잔액: ", balance)   #입금한 뒤 잔액을 가져옴
        print("거래해주셔서 감사합니다. - by Python Bank")

    def get_receipt2(self, amount, balance): #출금을 위한 명세표
        print("*"*25)
        print("{0:=^26}".format("명세표"))
        print("거래시간: ", time)
        print("이름: ", self.name)
        print("출금액: ", amount)
        print("남은 잔액: ",balance)
        print("거래해주셔서 감사합니다. - by Python Bank")
    def reg_customer(self):         #엑셀에 사용자 추가 하는 함수
        passwd = int(input(self.name + "님의 비밀번호를 입력하세요:
"))
        new_balance = int(input(self.name + "님의 초기잔액을
입력하세요: "))
        self.balance = new_balance

        #엑셀에 사용자 추가
        max_row = 7
        newCell_obj = sheet_obj['A' + str(max_row+1)]
        newCell_obj.value = self.name
        newCell_obj = sheet_obj['B' + str(max_row+1)]
        newCell_obj.value = passwd
        newCell2_obj = sheet_obj['C' + str(max_row+1)]
        newCell2_obj.value = new_balance
```

```
        wb_obj.save("UserList.xlsx")
        print("등록이 완료되었습니다!")
        max_row = max_row+1
```

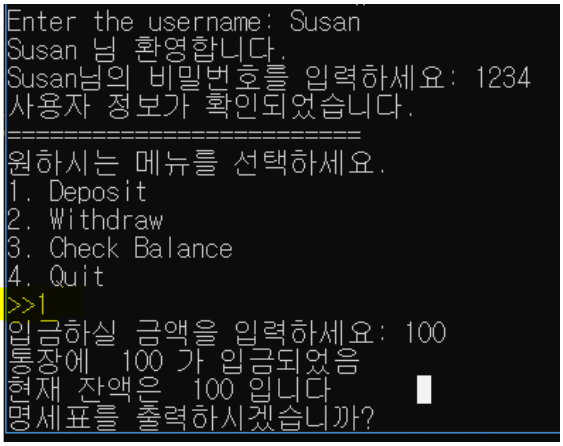
3. 단계별 결과와 캡쳐화면

A. 등록된 사용자인 경우

i. 이름-비밀번호 입력: 비밀번호가 일치하면 거래가 시작된다.



ii. Deposit(입금): 1을 눌러 입금을 한다. 현재 잔액이 0인 Susan은 입금 후 현재 잔액이 100으로 바뀐다.



iii. 입금 후 명세표 출력: yes라고 하면 명세표가 출력되고 다음 거래를 시작할 수 있다.

```
입금하실 금액을 입력하세요: 100
통장에  100 가 입금되었음
현재 잔액은  100 입니다
명세표를 출력하시겠습니까? yes
************************
          명세표
거래시간:  2019-06-12 18:17:07.311826
이름:  Susan
입금액:  100
남은 잔액:  100
거래해주셔서 감사합니다. - by Python Bank
========================
원하시는 메뉴를 선택하세요.
1. Deposit
2. Withdraw
3. Check Balance
4. Quit
>>
```

iv. Withdraw(출금): 2를 눌러 출금을 시작한다. 100에서 50을 출금했으니 남은 잔액은 50이다.

```
거래해주셔서 감사합니다. - by Python Bank
========================
원하시는 메뉴를 선택하세요.
1. Deposit
2. Withdraw
3. Check Balance
4. Quit
>>2
출금하실 금액을 입력하세요: 50
100
통장에  50 가 출금되었음
현재 잔액은  50 입니다
명세표를 출력하시겠습니까?
```

v. 출금 후 명세표 출력: yes를 눌러 명세표를 출력한다.

```
100
통장에  50 가 출금되었음
현재 잔액은  50 입니다
명세표를 출력하시겠습니까? yes
************************
==========명세표==========
거래시간:  2019-06-12 18:17:07.311826
이름:  Susan
출금액:  50
남은 잔액:  50
거래해주셔서 감사합니다. - by Python Bank
========================
원하시는 메뉴를 선택하세요.
1. Deposit
2. Withdraw
3. Check Balance
4. Quit
>>
```

vi. Check Balance(잔액 확인): 3을 눌러 현재잔액을 확인한다.

```
==========
원하시는 메뉴를 선택하세요.
1. Deposit
2. Withdraw
3. Check Balance
4. Quit
>>3
현재 잔액은 50 입니다.
==========
원하시는 메뉴를 선택하세요.
1. Deposit
2. Withdraw
3. Check Balance
4. Quit
>>
```

vii. Quit: 4를 누르면 프로그램이 종료가 된다.

```
==========
원하시는 메뉴를 선택하세요.
1. Deposit
2. Withdraw
3. Check Balance
4. Quit
>>4
>>>
```

B. 등록된 사용자가 아닌 경우

i. 이름입력: 이름을 입력했을 때 엑셀파일을 읽어와 등록된 사용자인지 검사한다.

```
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'classAtm'
>>> import classATM
>>> user2 = classATM.Atm()
Enter the username: Kim
Kim님은 등록되지 않았습니다. 추가하시겠습니까?
```

<현재 UserList에 등록된 사용자>

	A	B	C	D
1	username	pw	balance	
2	Susan	1234	0	
3	Mike	7251	50	
4	Nathan	2368	500	
5	Dan	4371	300	
6	John	4290	200	
7	Kevin	6103	150	
8				
9				

ii. 새로운 사용자 등록: 비밀번호와 초기잔액을 입력하면 엑셀파일에 자동으로 이름이 등록되고 거래를 진행할 수 있다.

```
>>> import classATM
>>> user2 = classATM.Atm()
Enter the username: Kim
Kim님은 등록되지 않았습니다. 추가하시겠습니까?yes
Kim님의 비밀번호를 입력하세요: 0000
Kim님의 초기잔액을 입력하세요: 500
등록이 완료되었습니다!
==========================
원하시는 메뉴를 선택하세요.
1. Deposit
2. Withdraw
3. Check Balance
4. Quit
>>
```

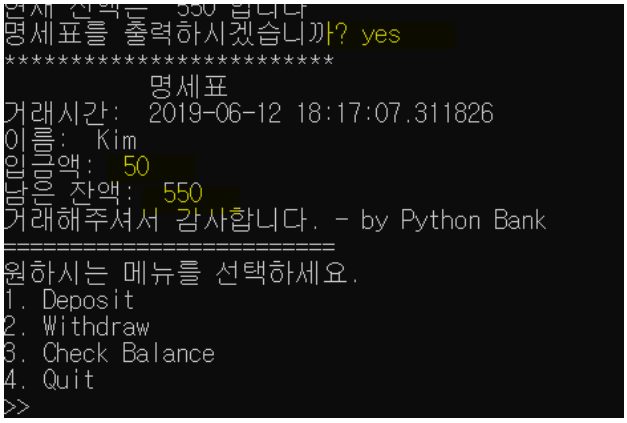
<액셀 파일 새로운 등록자 확인>

	A	B	C	D
1	username	pw	balance	
2	Susan	1234	0	
3	Mike	7251	50	
4	Nathan	2368	500	
5	Dan	4371	300	
6	John	4290	200	
7	Kevin	6103	150	
8	Kim	0	500	
9				

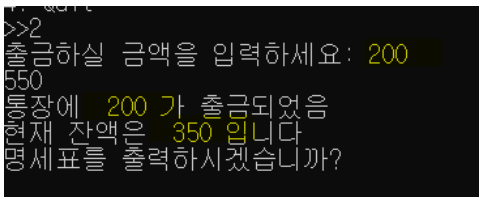
iii. Deposit(입금): 현재 잔액이 500인 Kim은 50을 입금하여 현재잔액이 550이 되었다.

```
Kim님의 초기잔액을 입력하세요: 500
등록이 완료되었습니다!
==========================
원하시는 메뉴를 선택하세요.
1. Deposit
2. Withdraw
3. Check Balance
4. Quit
>>1
입금하실 금액을 입력하세요: 50
통장에  50 가 입금되었음
현재 잔액은  550 입니다
명세표를 출력하시겠습니까?
```

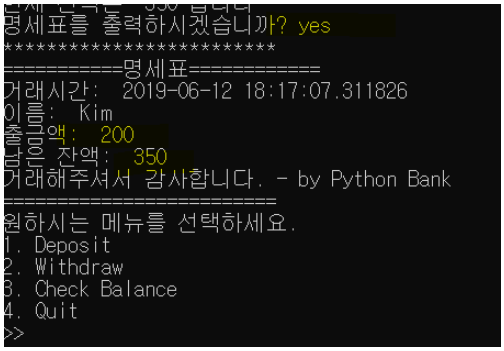
iv. 입금 후 명세표 출력: yes라고 하면 명세표가 출력된다.



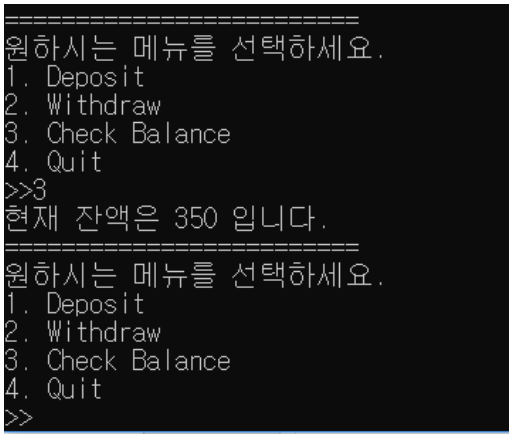
v. Withdraw(출금): 총 550에서 200을 출금하여 현재잔액은 350 이다.



vi. 출금 후 명세표 출력: yes를 눌러 명세표를 출력한다.



vii. Check Balance(잔액 확인)



viii. Quit: 4를 눌러 프로그램을 종료한다.