

Lab09

2022-1

소프트웨어의 이해

조교 시하연

hy.shi@sookmyung.ac.kr



#1. 은행계좌 관리 프로그램 - bank.py

- Account 클래스를 활용하여 계좌생성, 입금, 출금, 잔액조회 등의 기능을 갖는 은행계좌 관리 프로그램을 작성하시오.
- 여러 개의 계좌는 딕셔너리를 사용하여 저장하고 검색할 수 있다. (key : 계좌 번호 - value : Account 객체)

[메뉴 구성] - 메뉴를 선택하면 필요한 정보를 묻고 해당 작업 수행 후 결과 출력, 다시 메뉴로 돌아가서 반복

1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료

- Account 클래스에 구현해야 하는 메소드
 - 생성자 : 소유자 이름, 계좌 번호, 초기 금액을 매개변수로 받아 필드변수들을 설정, history 라는 이름의 필드변수를 빈 리스트로 설정하여 만들고, history 리스트에 신규 계좌 생성이라는 거래 내역 히스토리 추가 -> `append(("신규", 금액, 잔액))`
 - ✓ 모든 필드 변수는 `private`으로 구현할 것
 - 잔액 조회 : `__str__` 메소드를 활용하여 `print(계좌 객체)` 라고 작성하면 계좌 소유주, 계좌 번호, 잔액을 출력하도록 한다.
 - 입금 : 금액을 매개변수로 받아 입금하는 기능, 입금 히스토리 추가 -> `append(("입금", 금액, 잔액))`
 - 출금 : 금액을 매개변수로 받아 출금하는 기능, 출금 히스토리 추가 -> `append(("출금", 금액, 잔액))`
 - 거래 내역 조회 : history 리스트의 내용(해당 계좌의 거래 내역)을 출력

1. 신규 계좌 생성

- Account 객체를 생성하기 위해 사용자에게 계좌 소유자의 이름과, 계좌번호를 문자열로 입력 받는다.
- 여러 개의 계좌가 들어있는 딕셔너리에 입력 받은 계좌 번호 key가 있는지 확인하고 있으면 중복된 계좌 번호이므로 다시 입력하라는 메시지를 출력하고 계좌 번호를 다시 입력 받는다.
- 입력한 계좌 번호가 기존 계좌번호와 중복되는 것이 아니라면 입금할 금액을 입력 받고 Account 객체를 생성한다.
- 생성한 객체를 딕셔너리에 추가한다.
- 생성된 계좌 정보를 print(계좌 객체) 를 이용하여 출력한다.

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 1
계좌 소유자 이름을 입력하세요: 소웨이
생성할 계좌 번호를 입력하세요: 1234567
입금할 금액을 입력하세요: 100000

[ 생성된 계좌 정보 ]
계좌번호 : 1234567 , 소유자 : 소웨이 , 잔액 = 100000
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 1
계좌 소유자 이름을 입력하세요: 파이썬
생성할 계좌 번호를 입력하세요: 1234567
사용할 수 없는 계좌 번호입니다. 다시 입력하세요.
생성할 계좌 번호를 입력하세요: 1234567
사용할 수 없는 계좌 번호입니다. 다시 입력하세요.
생성할 계좌 번호를 입력하세요: 7654321
입금할 금액을 입력하세요: 50000

[ 생성된 계좌 정보 ]
계좌번호 : 7654321 , 소유자 : 파이썬 , 잔액 = 50000
```

2. 잔액 조회

- 사용자로부터 계좌 번호를 입력 받아 여러 개의 계좌가 들어있는 딕셔너리에 해당하는 계좌가 있다면 해당 계좌 객체를 출력하여 -> print(계좌 객체) 잔액 정보를 출력한다.
- 이때, 딕셔너리에 계좌가 없다면 해당 계좌가 없다는 메시지를 출력한다.

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료

선택 : 2
계좌 번호를 입력하세요 : 1234567
계좌번호 : 1234567 , 소유자 : 소웨이 , 잔액 = 100000
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료

선택 : 2
계좌 번호를 입력하세요 : 1111111
해당 계좌가 없습니다
```

3. 입금, 4. 출금

- 사용자로부터 계좌 번호를 입력 받아 여러 개의 계좌가 들어있는 딕셔너리에 해당하는 계좌가 있다면 해당 계좌 객체의 메소드로 입금, 출금 기능을 수행한다. 사용자로부터 금액을 입력 받아 메소드의 매개변수로 넣어 호출한다.
- 이때, 딕셔너리에 계좌가 없다면 해당 계좌가 없다는 메시지를 출력한다.

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 3
계좌 번호를 입력하세요 : 1234567
입금하실 금액을 입력하세요 : 20000
20000원이 입금되었습니다.
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 2
계좌 번호를 입력하세요 : 1234567
계좌번호 : 1234567 , 소유자 : 소웨이 , 잔액 = 120000
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 3
계좌 번호를 입력하세요 : 1111111
해당 계좌가 없습니다
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 4
계좌 번호를 입력하세요 : 7654321
출금하실 금액을 입력하세요 : 70000
잔액이 부족합니다.
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 4
계좌 번호를 입력하세요 : 7654321
출금하실 금액을 입력하세요 : 4000
4000원이 출금되었습니다.
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 2
계좌 번호를 입력하세요 : 7654321
계좌번호 : 7654321 , 소유자 : 파이썬 , 잔액 = 46000
```

5. 거래 내역 조회, 6. 종료

- 사용자로부터 계좌 번호를 입력 받아 여러 개의 계좌가 들어있는 딕셔너리에 해당하는 계좌가 있다면 해당 계좌 객체의 메소드를 이용하여 거래 내역 히스토리를 출력한다.
 - 이때, 딕셔너리에 계좌가 없다면 해당 계좌가 없다는 메시지를 출력한다.
 - 종료 메뉴 입력 시, 프로그램을 종료한다.
 - 1~6 이외의 정수 입력 시 다시 입력 받는다. 모든 메뉴 입력은 정수로 가정한다.
- ✓ 실습자료 예시 화면과 동일한 과정으로 입력하여 실행화면 모두 캡처
- ✓ (금액이나 이름 등은 조금씩 달라도 괜찮습니다.)

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 6
=====
이용해주셔서 감사합니다.
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 5
계좌 번호를 입력하세요 : 1234567
계좌번호 : 1234567 , 소유자 : 소웨이
신규    100000 원    잔액    100000 원
입금    20000 원    잔액    120000 원
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 5
계좌 번호를 입력하세요 : 7654321
계좌번호 : 7654321 , 소유자 : 파이썬
신규    50000 원    잔액    50000 원
출금    4000 원    잔액    46000 원
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입금
4. 출금
5. 거래 내역 조회
6. 종료
=====
선택 : 5
계좌 번호를 입력하세요 : 1111111
해당 계좌가 없습니다
```

과제 제출

- 과제 제출 기한
 - 실습 [다음주 수요일\(5/18\) 23시 00분](#)까지
- 제출 장소
 - [Snowboard](#) 해당 실습 과제 제출 페이지에 업로드

★ 표절검사!

과제 제출

- 과제를 수행하여 소스파일(.py)과 보고서(.pdf)를 한 폴더에 넣고,

“Lab과제번호_학번_이름.zip”으로 압축하여 제출

ex) Lab06_2201234_김눈송.zip

반드시 압축 파일 이름, 그 안에 파일 이름을 지켜 주시기 바랍니다.

- 소스파일 명: 매 실습마다 실습자료에 제시 예정

ex) triangle.py

- 보고서에 포함되어야 하는 항목 :

✓ 자기평가, 문제-기능별 실행화면 캡처, 소스코드 텍스트

문제	평가 항목	자기평가
1	신규 계좌 생성을 조건에 맞게 구현하였다.	/20
2	잔액 조회를 조건에 맞게 구현하였다.	/20
3	입출금을 조건에 맞게 구현하였다.	/20
4	거래 내역 조회를 조건에 맞게 구현하였다.	/20
5	Account 클래스의 모든 필드 변수를 private으로 구현하여 클래스 외부에서 필드 변수에 접근하지 않는 코드를 작성하였다.	/20
총점		/100

궁금한 점 + 질문



조교 이메일

hy.shi@sookmyung.ac.kr

[질문 시 주의!]

- ✓ 온라인 실습 시, 질문이 생길 경우에는 [Snowboard](#) 의 Q&A 게시판에 글 작성
- ✓ Q&A 게시판에는 그냥 코드만 올리면 (X)
- ✓ 몇 번 과제에서, 어떤 부분이 막혔는지 or 어떤 과정 중에 있는지 설명과 함께
- ✓ 꼭 과목, 분반, 전공, 이름, 학번을 알려주세요
- ✓ 실습 제출마감 당일 및 주말에는 답장이 늦을 수 있다는 점 고려해주세요 :)
- ✓ Office Hour
 - 메일로 시간 약속 잡기