

## Lab08\_2312282\_임다희

### #1. 은행계좌 관리 프로그램

```
bank.py - C:\Users\ao108\OneDrive\바탕 화면\bank.py (3.11.2)
File Edit Format Run Options Window Help

class Account:
    def __init__(self, name, num, bal):
        self.__name = name
        self.__num = num
        self.__bal = bal
        self.__history = []
        self.__history.append(('신규', self.__bal, self.__bal))

    def check_acc(self):
        return('계좌번호 : '+self.__num+ ' , 소유자 : '+self.__name+ ' , 잔액 : '+str(self.__bal)+'\n')

    def deposit(self, amount):
        self.__bal += amount
        self.__history.append(('입금', amount, self.__bal))
        print(amount, '원이 입금되었습니다.\n')

    def withdraw(self, amount):
        if amount > self.__bal:
            print('잔액이 부족합니다.\n')
        else:
            self.__bal -= amount
            self.__history.append(('출금', amount, self.__bal))
            print(amount, '원이 출금되었습니다.\n')

    def show_his(self):
        for (a, b, c) in self.__history:
            print(a, '\t', b, '\t원\t잔액\t', c)

accounts = {}
while True:
    print('='*36, '\n\t\t\t\t\t', 'SOOKMYUNG BANK ATM')
    print('='*36, '\n1. 신규 계좌 생성\n2. 잔액 조회\n3. 입금\n4. 출금\n5. 거래 내역 조회\n6. 종료')
    print('='*36)
    pick = int(input('선택 : '))

    if pick == 1:
        A = input('계좌 소유자 이름을 입력하세요 : ')
        while True:
            B = input('생성할 계좌 번호를 입력하세요 : ')
            if B in accounts:
                print('사용할 수 없는 계좌 번호입니다. 다시 입력하세요.')
            else:
                break
        C = int(input('입금할 금액을 입력하세요 : '))
        print('\n[ 생성된 계좌 정보 ]\n'+ '계좌번호 : ', B, ', ', '소유자 : ', A, ', ', '잔액 : ', C, '\n')
        account = Account(A, B, C)
        accounts[B] = account

    elif pick == 2:
        B = input('계좌 번호를 입력하세요 : ')
        if B in accounts:
            print(accounts[B].check_acc())
        else:
            print('해당 계좌가 없습니다.\n')

    elif pick == 3:
        B = input('계좌 번호를 입력하세요 : ')
        if B in accounts:
            amount = int(input('입금하실 금액을 입력하세요 : '))
            accounts[B].deposit(amount)
        else:
            print('해당 계좌가 없습니다.\n')

    elif pick == 4:
        B = input('계좌 번호를 입력하세요 : ')
        if B in accounts:
            amount = int(input('출금하실 금액을 입력하세요 : '))
            accounts[B].withdraw(amount)

    elif pick == 5:
        B = input('계좌 번호를 입력하세요 : ')
        if B in accounts:
            accounts[B].show_his()

    elif pick == 6:
        break
```

```

class Account:
    def __init__(self,name,num,bal):
        self.__name=name
        self.__num=num
        self.__bal=bal
        self.__history=[]
        self.__history.append(('신규',self.__bal,self.__bal))

    def check_acc(self):
        return('계좌번호 : '+self.__num+ ' , 소유자 : '+self.__name+ ' , 잔액 : '+str(self.__bal)+'\n')

    def deposit(self,amount):
        self.__bal+=amount
        self.__history.append(('입금',amount,self.__bal))
        print(amount,'원이 입금되었습니다.\n')

    def withdraw(self,amount):
        if amount>self.__bal:
            print('잔액이 부족합니다.\n')
        else:
            self.__bal-=amount
            self.__history.append(('출금',amount,self.__bal))
            print(amount,'원이 출금되었습니다.\n')

    def show_his(self):
        for (a,b,c) in self.__history:
            print(a,'\t',b,'원\t잔액\t',c)

accounts={}
while True:
    print('='*36,'\n'\t','SOOKMYUNG BANK ATM')
    print('='*36,'\n1.신규 계좌 생성\n2.잔액 조회\n3.입금\n4.출금\n5.거래 내역 조회\n6.종료')
    print('-'*36)
    pick=int(input('선택 : '))

    if pick==1:
        A=input('계좌 소유자 이름을 입력하세요: ')
        while True:
            B=input('생성할 계좌 번호를 입력하세요: ')
            if B in accounts:
                print('사용할 수 없는 계좌 번호입니다. 다시 입력하세요.')
            else:
                break
        C=int(input('입금할 금액을 입력하세요: '))
        print('\n[ 생성된 계좌 정보 ]\n'+계좌번호 : '.B.', 소유자 : '.A.', 잔액 : '.C.',\n')
        account=Account(A,B,C)
        accounts[B]=account

    elif pick==2:
        B=input('계좌 번호를 입력하세요: ')
        if B in accounts:
            print(accounts[B].check_acc())
        else:
            print('해당 계좌가 없습니다\n')

    elif pick==3:
        B=input('계좌 번호를 입력하세요: ')
        if B in accounts:
            amount=int(input('입금하실 금액을 입력하세요: '))
            accounts[B].deposit(amount)
        else:
            print('해당 계좌가 없습니다\n')

    elif pick==4:
        B=input('계좌 번호를 입력하세요: ')
        if B in accounts:
            amount=int(input('출금하실 금액을 입력하세요: '))
            accounts[B].withdraw(amount)

    elif pick==5:
        B=input('계좌 번호를 입력하세요: ')
        if B in accounts:
            accounts[B].show_his()

    elif pick==6:
        break

```

## 1. 신규 계좌 생성

The screenshot displays a Python IDE with two windows. The left window shows the source code for a bank account simulation, and the right window shows the execution output.

**Source Code (Left Window):**

```
bank.py - C:\Users\ja0108\OneDrive\바탕 화면\bank.py [3.11.2]
File Edit Format Run Options Window Help

class Account:
    def __init__(self,name,num,bal):
        self.__name=name
        self.__num=num
        self.__bal=bal
        self.__history=[]
        self.__history.append(('신규',self.__bal,self.__bal))

    def check_acc(self):
        return('계좌번호 : '+self.__num+' , 소유자 : '+self.__name+' , 잔액 : '+str(self.__bal)+'\n')

    def deposit(self,amount):
        self.__bal+=amount
        self.__history.append(('입금',amount,self.__bal))
        print(amount,'원이 입금되었습니다.\n')

    def withdraw(self,amount):
        if amount>self.__bal:
            print('잔액이 부족합니다.\n')
        else:
            self.__bal-=amount
            self.__history.append(('출금',amount,self.__bal))
            print(amount,'원이 출금되었습니다.\n')

    def show_his(self):
        for (a,b,c) in self.__history:
            print(a,'\t','b.','원\t잔액\t','c)

accounts={}
while True:
    print('='*36,'\n'+'\t','SOOKMYUNG BANK ATM')
    print('='*36,'\n1.신규 계좌 생성\n2.잔액 조회\n3.입금\n4.출금\n5.거래 내역 조회\n6.종료')
    pick=int(input('선택 : '))

    if pick==1:
        A=input('계좌 소유자 이름을 입력하세요 : ')
        while True:
            B=input('생성할 계좌 번호를 입력하세요 : ')
            if B in accounts:
                print('사용할 수 없는 계좌 번호입니다. 다시 입력하세요.')
            else:
                break
        C=int(input('입금할 금액을 입력하세요 : '))
        print('\n[ 생성된 계좌 정보 ]\n'+'\t계좌번호 : ',B,',', '소유자 : ',A,',', '잔액 : ',C,'\n')
        account=Account(A,B,C)
        accounts[B]=account

    elif pick==2:
        B=input('계좌 번호를 입력하세요 : ')
        if B in accounts:
            print(accounts[B].check_acc())
        else:
```

**Output (Right Window):**

```
*IDLE Shell 3.11.2*
File Edit Shell Debug Options Window Help

Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ja0108\OneDrive\바탕 화면\bank.py =====
=====
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
```

Ln: 55 Col: 5

## 2. 잔액 조회

The image shows a Python IDE with a file named 'bank.py' (3.11.2). The code defines a class 'Account' with methods for initialization, checking balance, depositing, withdrawing, and showing history. It also includes a main loop that creates an account and processes user input.

```

class Account:
    def __init__(self, name, num, bal):
        self.__name=name
        self.__num=num
        self.__bal=bal
        self.__history=[]
        self.__history.append(('신규',self.__bal,self.__bal))

    def check_acc(self):
        return('계좌번호 : '+self.__num+' , 소유자 : '+self.__name+' , 잔액 : '+str(self.__bal)+'\n')

    def deposit(self, amount):
        self.__bal+=amount
        self.__history.append(('입금',amount,self.__bal))
        print(amount, '원이 입금되었습니다.\n')

    def withdraw(self, amount):
        if amount>self.__bal:
            print('잔액이 부족합니다.\n')
        else:
            self.__bal-=amount
            self.__history.append(('출금',amount,self.__bal))
            print(amount, '원이 출금되었습니다.\n')

    def show_his(self):
        for (a,b,c) in self.__history:
            print(a, '원', b, '원', c)

accounts={}
while True:
    print('='*36+'\n'+'SOOKMYUNG BANK ATM')
    print('='*36+'\n'+'1.신규 계좌 생성 2.잔액 조회 3.입금 4.출금 5.거래 내역 조회 6.종료')
    print('='*36)
    pick=int(input('선택 : '))

```

The output window shows the execution of the program. It displays the menu, the user's selection of '2' (check balance), and the resulting account information: '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 : 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

```
bank.py - C:\Users\ja0108\OneDrive\바탕 화면\bank.py (3.11.2)
File Edit Format Run Options Window Help

class Account:
    def __init__(self, name, num, bal):
        self.__name=name
        self.__num=num
        self.__bal=bal
        self.__history=[]
        self.__history.append(('신규',self.__bal,self.__bal))

    def check_acc(self):
        return('계좌번호 : '+self.__num+ ' , 소유자 : '+self.__name+ ' , 잔액 : '+str(self.__bal))

    def deposit(self, amount):
        self.__bal+=amount
        self.__history.append(('입금',amount,self.__bal))
        print(amount, '원이 입금되었습니다.\n')

    def withdraw(self, amount):
        if amount>self.__bal:
            print('잔액이 부족합니다.\n')
        else:
            self.__bal-=amount
            self.__history.append(('출금',amount,self.__bal))
            print(amount, '원이 출금되었습니다.\n')

    def show_his(self):
        for (a,b,c) in self.__history:
            print(a, '원', b, '원', c)

accounts={}
while True:
    print('='*36, '\n' , 'SOOKMYUNG BANK ATM')
    print('='*36, '\n1.신규 계좌 생성\n2.잔액 조회\n3.입금\n4.출금\n5.거래 내역 조회\n6.종료')
    print('='*36)
    pick=int(input('선택 : '))

    if pick==1:
        A=input('계좌 번호를 입력하세요 : ')
        B=input('소유자 이름을 입력하세요 : ')
        while True:
            C=input('초기 금액을 입력하세요 : ')
            if C.isdigit():
                break
            else:
                print('숫자만 입력하세요.')
        accounts[A]=Account(B, A, int(C))
        print('계좌 생성 완료입니다.')
    elif pick==2:
        A=input('계좌 번호를 입력하세요 : ')
        if A in accounts:
            print(accounts[A].check_acc())
        else:
            print('계좌 번호가 없습니다.')
    elif pick==3:
        A=input('계좌 번호를 입력하세요 : ')
        B=input('입금 금액을 입력하세요 : ')
        while True:
            if B.isdigit():
                break
            else:
                print('숫자만 입력하세요.')
        accounts[A].deposit(int(B))
    elif pick==4:
        A=input('계좌 번호를 입력하세요 : ')
        B=input('출금 금액을 입력하세요 : ')
        while True:
            if B.isdigit():
                break
            else:
                print('숫자만 입력하세요.')
        accounts[A].withdraw(int(B))
    elif pick==5:
        A=input('계좌 번호를 입력하세요 : ')
        if A in accounts:
            accounts[A].show_his()
        else:
            print('계좌 번호가 없습니다.')
    elif pick==6:
        break
    else:
        print('잘못된 선택입니다.')

Ln: 123 Col: 5
```

### 4. 출금

```
bank.py - C:\Users\ja0108\OneDrive\바탕 화면\bank.py (3.11.2)
File Edit Format Run Options Window Help

class Account:
    def __init__(self, name, num, bal):
        self.__name=name
        self.__num=num
        self.__bal=bal
        self.__history=[]
        self.__history.append(('신규',self.__bal,self.__bal))

    def check_acc(self):
        return('계좌번호 : '+self.__num+ ' , 소유자 : '+self.__name+ ' , 잔액 : '+str(self.__bal))

    def deposit(self, amount):
        self.__bal+=amount
        self.__history.append(('입금',amount,self.__bal))
        print(amount, '원이 입금되었습니다.\n')

    def withdraw(self, amount):
        if amount>self.__bal:
            print('잔액이 부족합니다.\n')
        else:
            self.__bal-=amount
            self.__history.append(('출금',amount,self.__bal))
            print(amount, '원이 출금되었습니다.\n')

    def show_his(self):
        for (a,b,c) in self.__history:
            print(a, '원', b, '원', c)

accounts={}
while True:
    print('='*36, '\n' , 'SOOKMYUNG BANK ATM')
    print('='*36, '\n1.신규 계좌 생성\n2.잔액 조회\n3.입금\n4.출금\n5.거래 내역 조회\n6.종료')
    print('='*36)
    pick=int(input('선택 : '))

    if pick==1:
        A=input('계좌 번호를 입력하세요 : ')
        B=input('소유자 이름을 입력하세요 : ')
        while True:
            C=input('초기 금액을 입력하세요 : ')
            if C.isdigit():
                break
            else:
                print('숫자만 입력하세요.')
        accounts[A]=Account(B, A, int(C))
        print('계좌 생성 완료입니다.')
    elif pick==2:
        A=input('계좌 번호를 입력하세요 : ')
        if A in accounts:
            print(accounts[A].check_acc())
        else:
            print('계좌 번호가 없습니다.')
    elif pick==3:
        A=input('계좌 번호를 입력하세요 : ')
        B=input('입금 금액을 입력하세요 : ')
        while True:
            if B.isdigit():
                break
            else:
                print('숫자만 입력하세요.')
        accounts[A].deposit(int(B))
    elif pick==4:
        A=input('계좌 번호를 입력하세요 : ')
        B=input('출금 금액을 입력하세요 : ')
        while True:
            if B.isdigit():
                break
            else:
                print('숫자만 입력하세요.')
        accounts[A].withdraw(int(B))
    elif pick==5:
        A=input('계좌 번호를 입력하세요 : ')
        if A in accounts:
            accounts[A].show_his()
        else:
            print('계좌 번호가 없습니다.')
    elif pick==6:
        break
    else:
        print('잘못된 선택입니다.')

Ln: 153 Col: 5
```

## 5. 거래 내역 조회

```
bank.py - C:\Users\au108\OneDrive\바탕 화면\bank.py (3.11.2)
File Edit Format Run Options Window Help

class Account:
    def __init__(self, name, num, bal):
        self.__name = name
        self.__num = num
        self.__bal = bal
        self.__history = []
        self.__history.append(('신규', self.__bal, self.__bal))

    def check_acc(self):
        return('계좌번호 : ' + self.__num + ' , 소유자 : ' + self.__name + ' , 잔액 : ' + str(self.__bal) + '\n')

    def deposit(self, amount):
        self.__bal += amount
        self.__history.append(('입금', amount, self.__bal))
        print(amount, '원이 입금되었습니다.\n')

    def withdraw(self, amount):
        if amount > self.__bal:
            print('잔액이 부족합니다.\n')
        else:
            self.__bal -= amount
            self.__history.append(('출금', amount, self.__bal))
            print(amount, '원이 출금되었습니다.\n')

    def show_his(self):
        for (a, b, c) in self.__history:
            print(a, '\t', b, '\t', '원잔액\n', c)

accounts = {}
while True:
    print('=' * 36, '\n' * 2, 'SOOKMYUNG BANK ATM')
    print('=' * 36, '\n1. 신규 계좌 생성\n2. 잔액 조회\n3. 입출금\n4. 거래 내역 조회\n5. 종료')
    pick = int(input('선택 : '))

    if pick == 1:
        A = input('계좌 소유자 이름을 입력하세요 : ')
        while True:
            B = input('생성할 계좌 번호를 입력하세요 : ')
            if B in accounts:
                print('사용할 수 없는 계좌 번호입니다. 다시 입력하세요.')
            else:
                break
        C = int(input('입금할 금액을 입력하세요 : '))
        print('\n[ 생성된 계좌 정보 ]\n' + '계좌번호 : ' + B + ' , 소유자 : ' + A + ' , 잔액 : ' + C + '\n')
        account = Account(A, B, C)
        accounts[B] = account

    elif pick == 2:
        B = input('계좌 번호를 입력하세요 : ')
        if B in accounts:
            print(accounts[B].check_acc())
        else:
```

```
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입출금
4. 거래 내역 조회
5. 종료
선택 : 5
계좌 번호를 입력하세요 : 1234567
신규      100000 원      잔액      100000
입금      20000 원      잔액      80000
=====
SOOKMYUNG BANK ATM
=====
1. 신규 계좌 생성
2. 잔액 조회
3. 입출금
4. 거래 내역 조회
5. 종료
선택 : 5
계좌 번호를 입력하세요 : 7854321
신규      50000 원      잔액      50000
출금      4000 원      잔액      46000
=====
```

Ln: 181 Col: 5