Presentation templates with layout variations

JAVA
입출금 관리 프로그램

JAVA 프로젝트

소속 강남학원
이름 유명준

목차

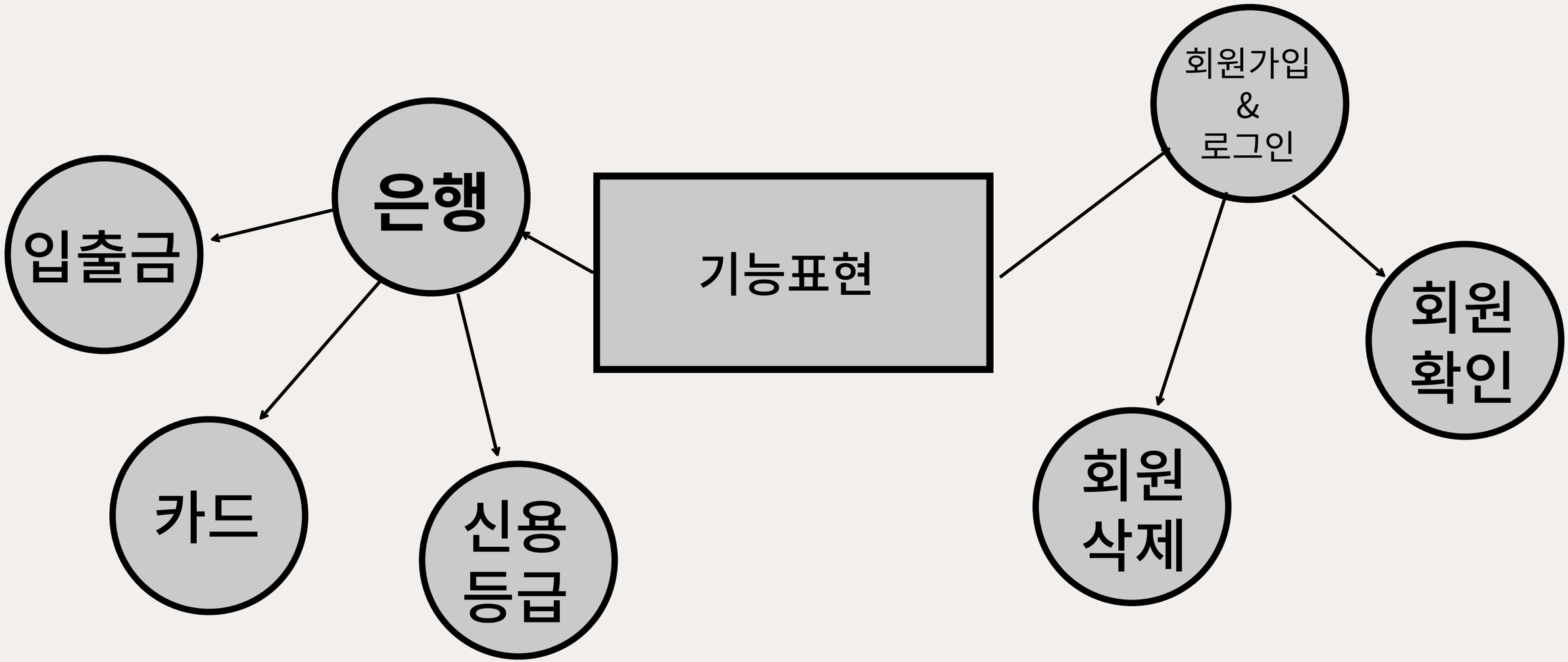
01 기획

주제 : 은행업무

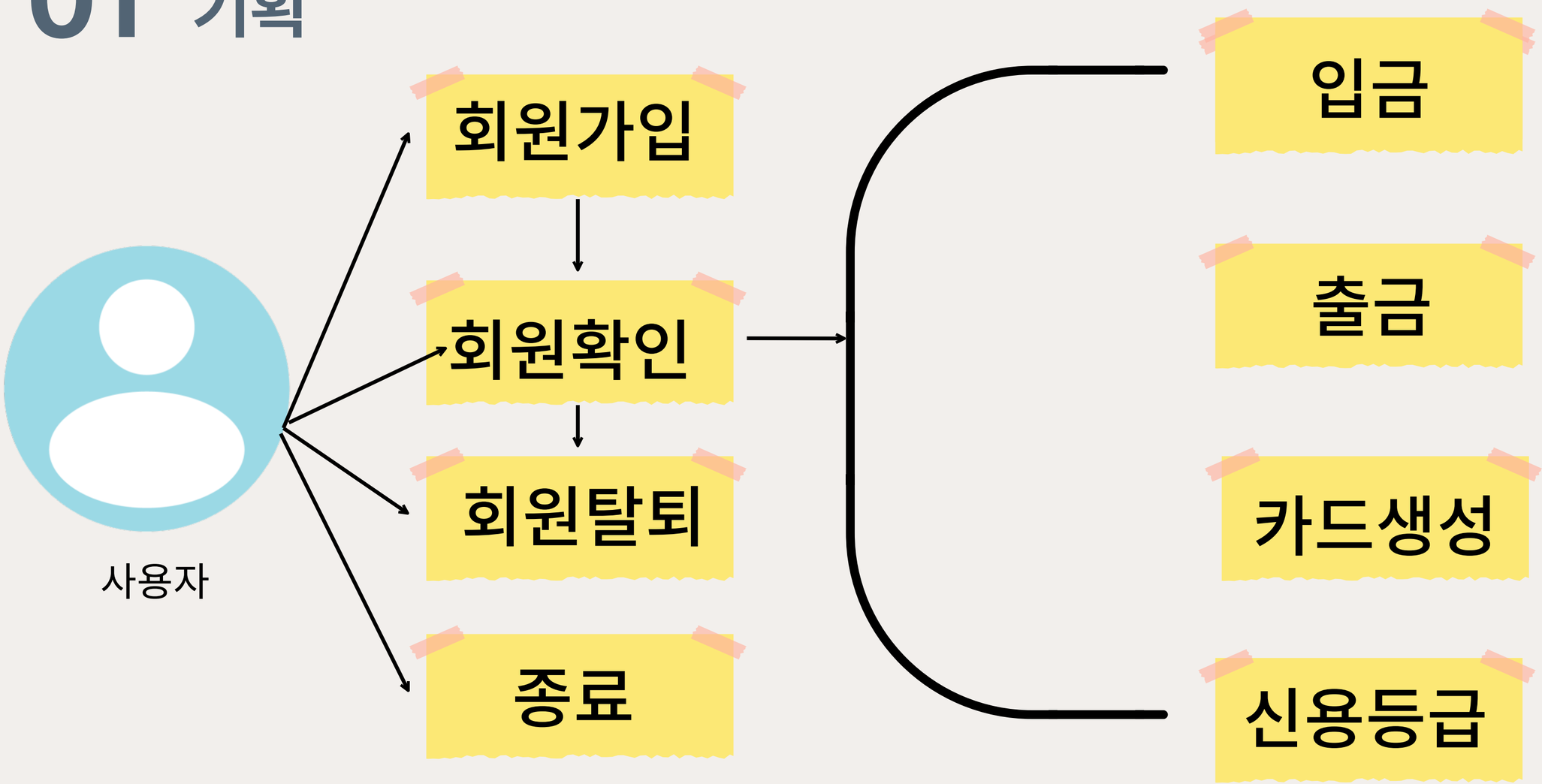
선정 이유: 수업내용으로 대부분 구현 가능하며
은행 분야 기업에 취업하고싶어서

이 템플릿은 텍스트를 쉽게 배치할 수 있는 레이아웃으로 구성되어 있습니다. 이 템플릿은 텍스트를 쉽게 배치할 수 있는 레이아웃으로 구성되어 있습니다. 이 템플릿은 텍스트를 쉽게 배치할 수 있는 레이아웃으로 구성되어 있습니다. 이 템플릿은 텍스트를 쉽게 배치할 수 있는 레이아웃으로 구성되어 있습니다.

01 기획



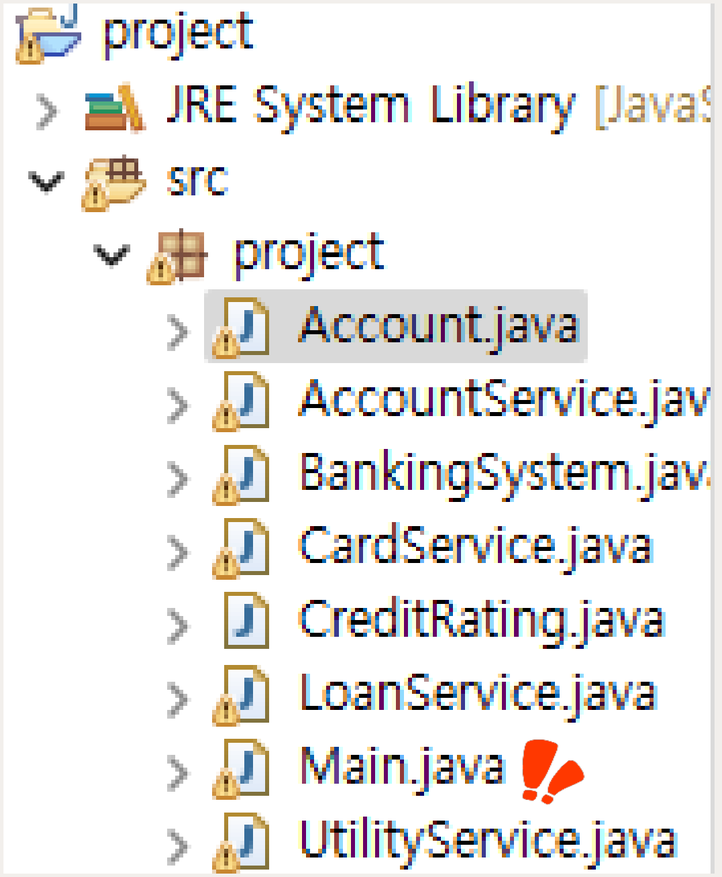
01 기획



02 디자인

아스키아트

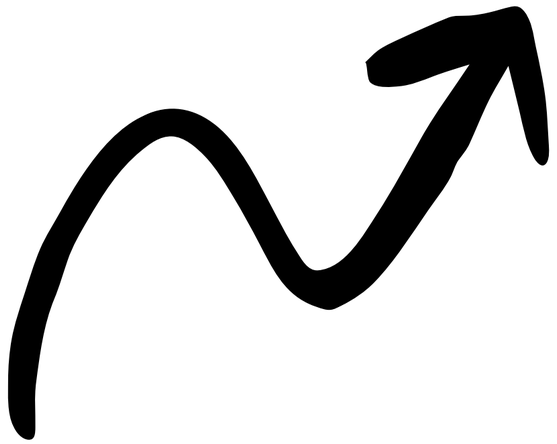
```
===== Java 은행 메뉴 시스템 =====
1.계좌서비스
2.은행시스템
3.카드서비스
4.신용등급 확인
5.계좌 삭제
0.종료
항목을 입력해주세요
```



8개의 클래스 구현

03 기능설명

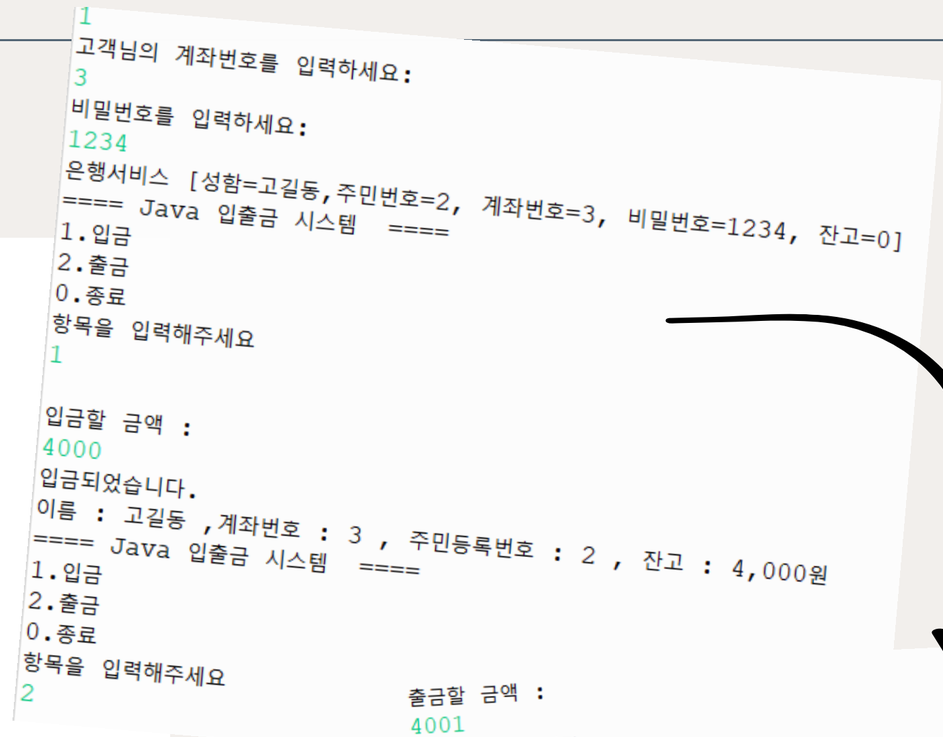
```
계좌개설을 실행합니다.
정확한 주민번호 앞번호 6자리, "-"를 붙이고 뒷번호 7자리를 입력해주세요
asd
올바른 주민번호를 입력해주세요
정확한 주민번호 앞번호 6자리, "-"를 붙이고 뒷번호 7자리를 입력해주세요
242252-5243122
계좌번호 15자리를 입력해주세요
123456789012345
이름을 입력해주세요
홍길동
비밀번호 4자리를 입력해주세요
1234
계좌가 생성되었습니다.
은행서비스 [성함=홍길동,주민번호=242252-5243122, 계좌번호=123456789012345, 비밀번호=1234, 잔고=0]
```

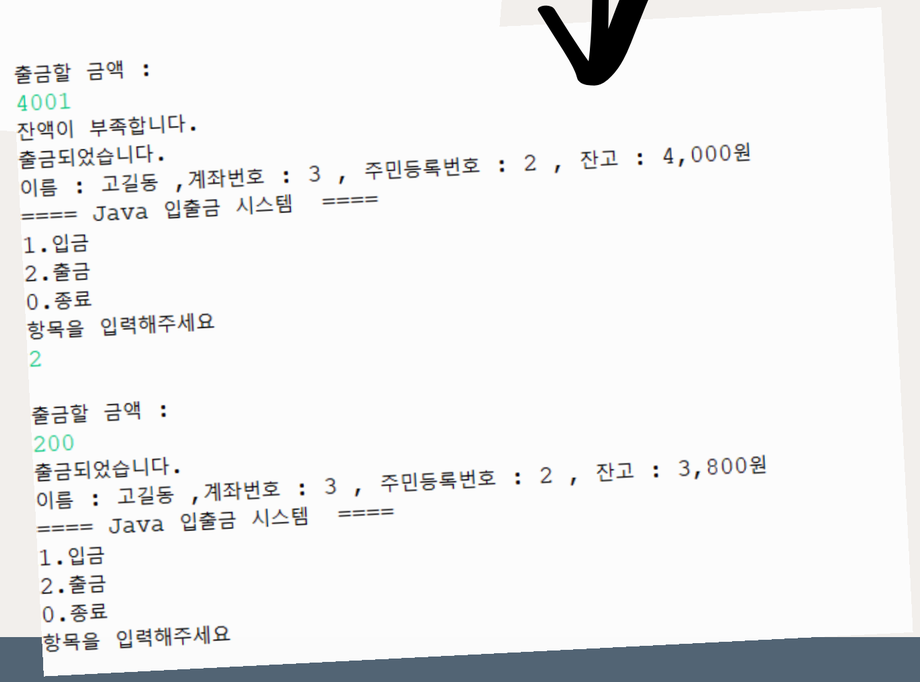


```
==== Java 은행 메뉴 시스템  ====
1.계좌서비스
2.은행시스템
3.카드서비스
4.신용등급 확인
5.계좌 삭제
0.종료
항목을 입력해주세요
2
==== 은행 서비스 관리 시스템 ====
1.계좌검색
2.모든 계좌
0.취소
2
```

03 기능설명

```
==== Java 은행 메뉴 시스템  ====
1.계좌서비스
2.은행시스템
3.카드서비스
4.신용등급 확인
5.계좌 삭제
0.종료
항목을 입력해주세요
2
==== 은행 서비스 관리 시스템 ====
1.계좌검색
2.모든 계좌
0.취소
2
모든 계좌를 출력합니다.
이름 : 홍길동 ,계좌번호 : 2 , 주민등록번호 : 1 , 잔고 : 0원
이름 : 고길동 ,계좌번호 : 3 , 주민등록번호 : 2 , 잔고 : 0원
이름 : 엄준식 ,계좌번호 : 4 , 주민등록번호 : 3 , 잔고 : 0원
==== 은행 서비스 관리 시스템 ====
1.계좌검색
2.모든 계좌
0.취소
```





임시로 입력값을 짧게 만듬

04 소스설명

```java
public class Main {
    public static void main(String[] args) {
// try-do-while 문을 만들어 해당 문장이 ture 값이 나올때까지 무한반복 순환문
// switch를 이용하여 해당 값이 나올때마다 로직이 이어지며 원하는 값이 나올때까지 반복
        try {
            boolean Exit = false;
            Scanner scanner = new Scanner(System.in);

            do {
                Scanner scan = new Scanner(System.in);
                System.out.println("==== Java 은행 메뉴 시스템  ====");
                System.out.println("1.계좌서비스");
                System.out.println("2.은행시스템");
                System.out.println("3.카드서비스");
                System.out.println("4.신용등급 확인");
                System.out.println("5.계좌 삭제");
                System.out.println("0.종료");
                System.out.println("항목을 입력해주세요");

                int menu;
                try {
                    menu = scan.nextInt();
                } catch (InputMismatchException e) {
                    System.out.println("잘못된 입력입니다. 숫자를 입력해주세요.");
                    scanner.next();
                    continue;
                }
                switch (menu) {
                case 1:
                    AccountService.CreateAccount();
                    break;

                case 0: // 종료
                    System.out.println("종료합니다.");
                    Exit = true;
                    scan.close();
                    break;
                }
                //순환문을 트루
            } while (!Exit);
        } catch (Exception e) {
            System.out.println("오류");
            e.printStackTrace();

        }
    }
```

04 소스설명

```
try {
    //통장 생성
    // 새로운 잔액 생성자 생성
    //socialNumber, accountNumber, name, pwd ,bin 값을 받고 객체배열에 넣으며
    //메서드가 반복될때마다 index값을 증감연산자로 1씩 카운트 올림
    int bin=0 ;
    Account account = new Account(socialNumber, accountNumber, name, pwd ,bin);
    member[index++]= account;
    System.out.println("계좌가 생성되었습니다.");
    System.out.println(account.Account_to()); // 계좌 정보 출력
} catch (Exception e) {
    System.out.println("에러발생");
    System.out.println(e.getMessage());
}
}
```

```
//모든 계좌 호출
//member[i]객체 배열에 값이 없을경우 printallAccount()호출
public static void all_Account() {
    for(int i =0 ; i<index; i++) {
        if (member[i] != null) {
            member[i].printallAccount();
        }
    }
}
```

04 소스 설명

```java
package project;
//계좌 서비스
import java.util.Scanner;

public class Account {
    //인스턴스 변수
    private  String SocialNumber;    //주민번호
    private  String accountNumber;   //계좌번호
    private String name;             //성함
    private String pwd;              //비밀번호
    private  int bin;                //잔고

    //디폴트
    public Account() {}
    //-----------------생성자--------------------
    public Account(String socialNumber, String accountNumber, Str
        super();
        SocialNumber = socialNumber;
        this.accountNumber = accountNumber;
        this.name = name;
        this.pwd = pwd;
        this.bin = bin;
    }
    public Account(String socialNumber, String accountNumber,
            String name, String pwd) {
        this(socialNumber,accountNumber,name,pwd,0);
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
```

인스턴스 5개로 구성 get-set tostring 이용

```java
//모든 계좌 호출
public  void printallAccount() {
    System.out.printf("이름 : %s ,계좌번호 : %s , 주민등록번호 : %s , 잔고 : %,d원%n"
                      ,name,accountNumber,SocialNumber,bin);
}
//단일 계좌 호출
public  String  Account_to() {
    return "은행서비스 [성함="+ name +
           ",주민번호=" + SocialNumber +
           ", 계좌번호=" + accountNumber +
           ", 비밀번호=" + pwd +
           ", 잔고=" + bin + "]";
}
```

04 소스설명

```
public class AccountService  {
    //객체 배열 지정
    static Account[] member = new Account[100];
    static int index = 0;

    public static void CreateAccount() {
        Scanner scanner = new Scanner(System.in);
        boolean cahm = false;

        String socialNumber = "";
        String accountNumber = "";
        String name = "";
        String pwd = "";


        System.out.println("계좌개설을 실행합니다.");

        // 주민번호 만들기
        while (!cahm) {
            cahm = false;
            System.out.println("정확한 주민번호 앞번호 6자리, \"-\"를 붙이고 뒷번호 7자리를 입력해주세요");
             socialNumber = scanner.nextLine();
            if (!Pattern.matches("^[0-9]{6}-[0-9]{7}$", socialNumber)) {
                System.out.println("올바른 주민번호를 입력해주세요");
                    // 입력값이 해당 주민번호값과 객체배열에 같은 값이 있는지 확인하는
                    // 변수
            }else if(SocialNumber_faund(socialNumber)){
                System.out.println("중복된 주민번호 입니다.");
            }
            else {
                cahm = true;
            }
```

```
//객체배열의 주민등록값을 입력값과 비교하는 메서드
private static boolean SocialNumber_faund(String socialNumber) {
    for (int i = 0; i < index; i++) {
        if (member[i] != null && member[i].getSocialNumber()
                .equals(socialNumber)) {
            return true;
        }
    }
    return false;
}
```

04 소스설명 원하는 객체 배열 값 찾기

```
// 해당 계좌를 찾는 메서드
    public static void find_Account() {
        Scanner sc = new Scanner(System.in);
        boolean Exit = false;

        System.out.println("고객님의 계좌번호를 입력하세요: ");
        String accountNumber = sc.nextLine();

        System.out.println("비밀번호를 입력하세요: ");
        String pwd = sc.nextLine();

//accountNumber,pwd 입력값을 받고 find(accountNumber,pwd)메서드를 실행
        Account account = find(accountNumber,pwd);
        if(account == null) {
            System.out.println("계좌번호 또는 비밀번호를 정확히 입력해주세요");
            return;
        }
```

04 소스설명

원하는 객체 배열 값 찾기

```
//입력값과 member[i] 객체에 해당 값이 일치한지 순회 찾을경우 해당 member[i]인덱스를 반환
public static Account find(String AccountNumber,String pwd) {
    for(int i=0; i< index; i++) {
    if (member[i].getAccountNumber().equals(AccountNumber) &&
            member[i].getPwd().equals(pwd)) {
        System.out.println(member[i].Account_to());
        return member[i]; // 계좌를 찾았으면 인덱스 반환
    }
}
    return null;
}
```

04 소스설명

```
case 1: // 입금
        Scanner scan8 = new Scanner(System.in);

        System.out.println("입금할 금액 : ");
        try {
        int amount =Integer.parseInt(scan8.nextLine());
        account.deposit(amount);
        System.out.println("입금되었습니다.");
        account.printallAccount();
        } catch (Exception e) {
            System.out.println("유효한 정수를 입력하세요.");
        }
    break;

case 2:// 출금

    Scanner scan7 = new Scanner(System.in);
    System.out.println("출금할 금액 : ");
    try {
    int amount2 =Integer.parseInt(scan7.nextLine());
    account.minus2(amount2);
    System.out.println("출금되었습니다.");
    account.printallAccount();
    } catch (Exception e) {
        System.out.println("유효한 정수를 입력하세요.");
    }
    break;
case 0: // 종료
```

```
public  void deposit(int amount) {
bin += Math.abs(amount);
// 입출금에 입력값을 음수값으로 줬을때 대비


}
public  void minus2(int amount2) {
// 잔액<= 0일때 대비
if(bin < Math.abs(amount2)) {
    System.out.println("잔액이 부족합니다.");
    return;
}else {
    bin -= Math.abs(amount2);
}


}
```

05 시연

유명준

https://velog.io/@upjp/posts

감사합니다