프로젝트명 : 은행 고객들의 계좌관리 프로그램

4조 김은산, 김예니, 원혜민

도메인 클래스 (1)

```java
public class Account {
    private String accountId;
    private Customer customer;
    private double balance;

    public Account(String accountId, Customer customer, double balance) {
        this.accountId = accountId;
        this.customer = customer;
        this.balance = balance;
    }

    public String getAccountId() {
        return accountId;
    }

    public Customer getCustomer() { return customer; }

    public double getBalance() { return balance; }

    public void deposit(double amount) { balance += amount; }
```

도메인 클래스 (2)

```java
public class Customer {
    private String customerId;
    private String name;

    public Customer(String customerId, String name) {
        this.customerId = customerId;
        this.name = name;
    }

    public String getCustomerId() {
        return customerId;
    }

    public String getName() {
        return name;
    }

    @Override
    public String toString() {
        return "Customer{" +
                "customerId='" + customerId + '\'' +
                ", name='" + name + '\'' +
                '}';
    }
```

인터페이스 클래스

```java
package com.team.sec01;

public interface AccountManagement {
    void addAccount(Account account);
    void printAllAccounts();
    Account findAccountById(String accountId);
    void deposit(String accountId, double amount);
    void withdraw(String accountId, double amount);
}
```

실행 클래스

```
5  public class BankManagementSystem implements AccountManagement {
6      private ArrayList<Account> accounts;
7      private ArrayList<Customer> customers;
8
9      public BankManagementSystem() {
10         accounts = new ArrayList<>();
11         customers = new ArrayList<>();
12     }
13
14     @Override
15     public void addAccount(Account account) {
16         accounts.add(account);
17     }
18
19     @Override
20     public void printAllAccounts() {
21         for (Account account : accounts) {
22             System.out.println(account);
23         }
24     }
25
```

```java
    @Override
    public void deposit(String accountId, double amount) {
        Account account = findAccountById(accountId);
        if (account != null) {
            account.deposit(amount);
            System.out.println("입금 완료");
        } else {
            System.out.println("해당 계좌를 찾을 수 없습니다.");
        }
    }

    @Override
    public void withdraw(String accountId, double amount) {
        Account account = findAccountById(accountId);
        if (account != null) {
            account.withdraw(amount);
            System.out.println("출금 완료");
        } else {
            System.out.println("해당 계좌를 찾을 수 없습니다.");
        }
    }

    public void addCustomer(Customer customer) {
        customers.add(customer);
    }
```

- 메인 메소드

```
62      public static void main(String[] args) {
63          BankManagementSystem bank = new BankManagementSystem();
64          Scanner scanner = new Scanner(System.in);
65
66          while (true) {
67              System.out.println("1. 계좌 생성");
68              System.out.println("2. 모든 계좌 출력");
69              System.out.println("3. 계좌 입금");
70              System.out.println("4. 계좌 출금");
71              System.out.println("5. 종료");
72              System.out.print("선택: ");
73              int choice = scanner.nextInt();
74              scanner.nextLine(); // 버퍼 비우기
```

```
switch (choice) {
    case 1:
        System.out.print("고객 ID: ");
        String customerId = scanner.nextLine();
        System.out.print("고객 이름: ");
        String customerName = scanner.nextLine();
        bank.addCustomer(new Customer(customerId, customerName));

        System.out.print("계좌 번호: ");
        String accountId = scanner.nextLine();
        System.out.print("잔액: ");
        double balance = scanner.nextDouble();
        bank.addAccount(new Account(accountId, new Customer(customerId, customerName), balance));
        break;
    case 2:
        bank.printAllAccounts();
        break;
    case 3:
        System.out.print("입금할 계좌 번호: ");
        String depositAccountId = scanner.nextLine();
        System.out.print("입금할 금액: ");
        double depositAmount = scanner.nextDouble();
        bank.deposit(depositAccountId, depositAmount);
        break;
```

```
100         case 4:
101             System.out.print("출금할 계좌 번호: ");
102             String withdrawAccountId = scanner.nextLine();
103             System.out.print("출금할 금액: ");
104             double withdrawAmount = scanner.nextDouble();
105             bank.withdraw(withdrawAccountId, withdrawAmount);
106             break;
107         case 5:
108             System.out.println("프로그램을 종료합니다.");
109             System.exit( status: 0);
110             break;
111         default:
112             System.out.println("잘못된 선택입니다. 다시 선택해주세요.");
113     }
114
```