JAVA&MSA 정기수행 평가

계좌 관리 프로그램

2024-01-19

수강생 최형욱

목차

형성평가 실습 개요

▶주제

키보드로부터 계좌 정보를 입력 받아 관리하는 프로그램

▶ 구현 설계

계좌는 Account 객체로 생성

Bank Application에서 List형식으로 관리

형성평가 실습

Account class

1. 계좌번호, 계좌주, 초기입금액 멤버를 생성

2. 초기화 생성자를 생성

3. 데이터를 불러오기, 저장하기를 하기위해 Getter, Setter방식 사용

```
public class Account {
    private String ano;
    private String owner;
    private int balance;
    public Account() {}  //기본 생성자 필수

    public Account(String ano, String owner, int balance) {
        this.ano = ano;
        this.owner = owner;
        this.balance = balance;
    }

    public String getAno() {
        return ano;
    }
    public void setAno(String ano) {
        this.ano = ano;
    }

    public String getOwner() {
        return owner;
    }

    public void setOwner(String owner) {
        this.owner = owner;
    }

    public int getBalance() {
        return balance;
    }

    public void setBalance(int balance) {
        this.balance = balance;
    }
}
```

형성평가 실습

Bank Application Class

1. Account Type의 List 객체 생성

2. 입력 받을 객체 생성

3. try-catch문을 사용하여 정수가 아닌 경우 예외를 처리

4. while로 run변수가 true인 동안 계속 실행

5.사용자가 5번을 선택하면 run변수를 false로 설정하여 루프를 종료

6. 잘못된 선택에 대한 처리를 위해 default사용

```
public class BankApplication {
    private static List<Account> accounts = new ArrayList<>();
    private static Scanner scanner = new Scanner(System.in);
    static int count = 0;

    public static void main(String[] args) {
        boolean run = true;
        // 반복문으로 계속 수행, 5번 입력시 루프빠져나감
        while (run) {
            System.out.println("---------------------------------------------------------");
            System.out.println("1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료");
            System.out.println("---------------------------------------------------------");
            System.out.print("선택> ");

            int menuNum;
            // 정수가 아닐시 예외처리
            try {
                menuNum = scanner.nextInt();
                scanner.nextLine();
            } catch (InputMismatchException e) {
                System.out.println("숫자를 입력하세요.");
                scanner.nextLine();
                continue;
            }
            //
            switch (menuNum) {
                case 1:
                    createAccount();
                    break;
                case 2:
                    accountList();
                    break;
                case 3:
                    deposit();
                    break;
                case 4:
                    withdraw();
                    break;
                case 5:
                    run = false;
                    break;
                default:
                    System.out.println("잘못된 선택입니다. 다시 선택해주세요.");

            }
        }
        System.out.println("프로그램 종료");
    }
```

형성평가 실습
Bank Application Class

Method 구현

1. 계좌 생성 Method - createAccount()

사용자로부터 계좌번호, 계좌주, 초기 입금액을 입력 받고 List객체에 저장. 계좌 중복 여부를 확인을 위해 boolean을 사용

```
// 계좌 생성하기
private static void createAccount() {
    System.out.println("-----------");
    System.out.println("   계좌생성 ");
    System.out.println("-----------");

    String cNum = "";
    // 루프를 사용하여 계좌번호 입력 받음
    while (true) {
        boolean bool = true;    // 계좌번호의 중복 여부를 확인하기 위한 변수 초기화
        System.out.print("계좌번호: ");
        // 사용자로부터 계좌번호 입력받음
        cNum = scanner.next();
        for (int i = 0; i < count; i++) {
            if (cNum.equals(accounts.get(i).getAno())) {
                bool = false;
                break;
            }
        }
        // 계좌가 번호 중복방지
        if (!bool) {    // 중복된 계좌번호가 있을 경우 false로 설정하고 계속 반복
            System.out.print("중복된 계좌입니다. ");
        } else if (bool) {
            break;
        }
    }
    System.out.print("계좌주: ");
    String cName = scanner.next();
    System.out.print("초기입금액: ");
    int cMoney = scanner.nextInt();

    Account a = new Account(cNum, cName, cMoney);
    add(a);

    System.out.println("결과: 계좌가 생성되었습니다.");
}
```

형성평가 실습
Bank Application Class

Method 구현

1. 계좌 목록 조회 Method - accountList()

For문을 사용하여 등록된 각 계좌에 대한 정보를 출력

Count를 사용하여 현재 등록된 계좌의 수

2. 계좌 예금 Method – deposit()

findAccount를 호출하여 예금을 수행할 계좌를 찾고 사용자로부터 계좌를 입력받고 해당 계좌를 찾아 반환하는 역할

```
//계좌목록조회
private static void accountList() {
    System.out.println("-----------");
    System.out.println("   계좌목록 ");
    System.out.println("-----------");
    for (int i = 0; i < count; i++) {
        System.out.print(accounts.get(i).getAno()+"\t");
        System.out.printf("%s", accounts.get(i).getOwner()+"\t");
        System.out.printf("%d", accounts.get(i).getBalance());
        System.out.println("");
    }
}

// 예금하기
private static void deposit() {

    Account mine = findAccount("예금");
    int amount = scanner.nextInt();
    mine.setBalance(mine.getBalance() + amount);
    System.out.println();

}
```

형성평가 실습

Bank Application Class

Method 구현

1. 계좌 출금 method - withdraw()

findAccount를 호출하여 출금을 수행할 계좌를 조회, 찾은 계좌는 myAccount 변수에 저장

음수처리와 출금 가능여부 확인 처리

2. 계좌조회 method – findAccount()

계좌번호 입력 및 조회를, 계봐번호 조회, 계좌 조회 성공 처리

Return을 통해 찾은 계좌의 정보를 반환

3. 리스트추가 method – add()

Count++를 통해 전체계좌 수를 증가

```
    private static void withdraw() {
        Account myAccount = findAccount("출금");
        int amount = scanner.nextInt();

        if (amount < 0) {    // 입력값이 음수일시
            System.out.println("잘못된 입력입니다.");
            return;
        }

         if (myAccount.getBalance() >= amount) {
            myAccount.setBalance(myAccount.getBalance() - amount);
            System.out.println("결과: 출금이 성공되었습니다.");
         } else {    // 잔액을 초과한 경우
            System.out.println("잔액이 부족합니다.");
         }
         System.out.println();
    }
    // 계좌 조회
    private static Account findAccount(String ano) {
        int myNum = 0;
        boolean bool = true;
        boolean bb = true;

        System.out.println("------------");
        System.out.println("    " + ano  );
        System.out.println("------------");

        while (bb) {
            System.out.print("계좌번호입력>> ");
            String myAcc = scanner.next();

            for (int i = 0; i < count; i++) {
                if (myAcc.equals(accounts.get(i).getAno())) {
                    myNum = i;
                    bool = false;
                    break;
                }
            }
            if (bool) {
                System.out.println("계좌번호 조회 실패했습니다. 다시 입력하세요");
            } else if (!bool) {
                System.out.print(ano + "액: ");
                bb = false;
            }
        }
        return accounts.get(myNum);
    }
    // 새로운 계좌를 리스트에 추가
    private static void add(Account account) {
        accounts.add(account);
        count++;
    }
}
```

구현결과

```
------------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------------------
선택> 1
-----------
   계좌생성
-----------
계좌번호: 111-111
계좌주: 홍길동
초기입금액: 10000
결과: 계좌가 생성되었습니다.
------------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------------------
선택> 1
-----------
   계좌생성
-----------
계좌번호: 111-222
계좌주: 강자바
초기입금액: 20000
결과: 계좌가 생성되었습니다.
------------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------------------
```

```
------------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------------------
선택> 2
-----------
   계좌목록
-----------
111-111 홍길동    10000
111-222 강자바    20000


------------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------------------
선택> 3
-----------
   예금
-----------
계좌번호입력>> 111-111
예금액: 5000
```

구현결과

```
------------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------------------
선택> 4
-----------
   출금
-----------
계좌번호입력>> 111-222
출금액: 3000
결과: 출금이 성공되었습니다.

------------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------------------
선택> 2
-----------
   계좌목록
-----------
111-111 홍길동    15000
111-222 강자바    17000
```

```
------------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------------------
선택> 5
프로그램 종료

------------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------------------
선택> -1
잘못된 선택입니다. 다시 선택해주세요.
------------------------------------------------------------
1.계좌생성 | 2.계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------------------
선택> ㅂㅈㄷ
숫자를 입력하세요.
```

감사합니다