|  |
| --- |
| 2017년 09월 26일 Java 실습보고서 |

|  |
| --- |
| 실습 2) BankManager – 객체의 이해 |

**■ 소스코드**

**[ex01.java]**

/\*

\* [실습2] BankManager

\* BankAccount를 관리하는 BankManager클래스를 구성하시오.

\*/

**package** source;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**import** java.util.Scanner;

**class** BankAccount{ // 계좌정보 클래스

**private** **int** accountNumber; // 계좌번호

**private** String customerName; // 고객명

**private** **double** accountBalance; // 잔액

**private** **static** **int** *count*; // 계좌(통장)발급 개수(객체 공유 변수)

**public** BankAccount(**int** accountNumber, String customerName){ // 매개변수 2개인 생성자

**this**(accountNumber, customerName, 0);

}

**public** BankAccount(**int** accountNumber, String customerName, **double** accountBalance){ // 매개변수 3개인 생성자

**this**.accountBalance=accountBalance;

**this**.customerName=customerName;

**this**.accountNumber=accountNumber;

*count*++; // 계좌 발급 개수 증가

}

**public** **static** String getCount(){ // count값 반환

**return** "[전체 계좌 수 : "+*count*+"]";

}

**public** **int** getNumber(){

**return** **this**.accountNumber;

}

**public** String getName(){

**return** **this**.customerName;

}

**public** **double** getBalance(){

**return** **this**.accountBalance;

}

**public** **void** deposit(**double** money){ // 입금

**this**.accountBalance+=money;

}

**public** **void** withdraw(**double** money){ // 출금

**if**(**this**.accountBalance>=money)

**this**.accountBalance-=money;

**else**

System.***out***.println("잔액이 부족합니다.");

}

**public** **void** transfer(BankAccount acct, **double** money){ // 계좌이체

**if**(**this**.accountBalance>=money){

**this**.withdraw(money);

acct.deposit(money); // 참조객체

}**else**

System.***out***.println("잔액이 부족합니다.");

}

**public** String toString(){ // 문자열 만들어서 리턴

String customerInfo="계좌번호 : "+**this**.accountNumber+", 고객명 : "

+**this**.customerName+", "+"잔액 : "+**this**.accountBalance;

**return** customerInfo;

}

}

**public** **class** ex01 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

*printName*();

BankManager bank=**new** BankManager();

bank.create();

bank.create();

bank.create();

bank.deposit();

bank.transfer();

System.***out***.println(bank.toString());

}

}

**class** BankManager{ // BankAccount클래스 관리

**private** BankAccount[] accounts;

**private** **static** **final** **int** ***num***=20;

**private** **int** count=0;

Scanner sc=**new** Scanner(System.***in***);

**public** BankManager(){

accounts=**new** BankAccount[***num***];

}

**public** **void** create(){ // 계좌개설

System.***out***.println("[계좌개설]");

System.***out***.print("계좌 번호 입력 : ");

**int** account=sc.nextInt();

**if**(findAccount(account)==-1){ // 계좌번호가 없을 경우

System.***out***.print("이름 : ");

String name=sc.next();

System.***out***.print("입금액 : ");

**double** money=sc.nextDouble();

accounts[count++]=**new** BankAccount(account, name, money);

}**else**

System.***out***.println("계좌번호가 중복됩니다.");

}

**public** **void** deposit(){ // 입금

System.***out***.println("[입금]");

System.***out***.print("계좌 번호 입력 : ");

**int** account=sc.nextInt();

**int** index=findAccount(account);

**if**(index!=-1){ // 계좌 존재하는 경우

System.***out***.print("입금액 : ");

**double** money=sc.nextDouble();

accounts[index].deposit(money);

}

}

**public** **void** withdraw(){ // 출금

System.***out***.println("[출금]");

System.***out***.print("계좌 번호 입력 : ");

**int** account=sc.nextInt();

**int** index=findAccount(account);

**if**(index!=-1){

System.***out***.print("출금액 : ");

**double** money=sc.nextDouble();

accounts[index].withdraw(money);

}

}

**public** **void** transfer(){ // 이체

System.***out***.println("[계좌이체]");

System.***out***.print("계좌 번호 입력 : ");

**int** account=sc.nextInt();

**int** index=findAccount(account);

**if**(index!=-1){

System.***out***.print("입금할 계좌번호 입력 : ");

**int** account2=sc.nextInt();

**int** index2=findAccount(account2);

**if**(index2!=-1){

System.***out***.print("금액 입력 : ");

**double** money=sc.nextDouble();

accounts[index].transfer(accounts[index2], money);

}

}

}

**public** **int** findAccount(**int** target){ // 계좌번호 검색

**for**(**int** i=0; i<count; i++){

**if**(accounts[i].getNumber()==target)

**return** i;

}

**return** -1;

}

**public** String toString(){

String accInfo=**new** String();

**for**(**int** i=0; i<count; i++)

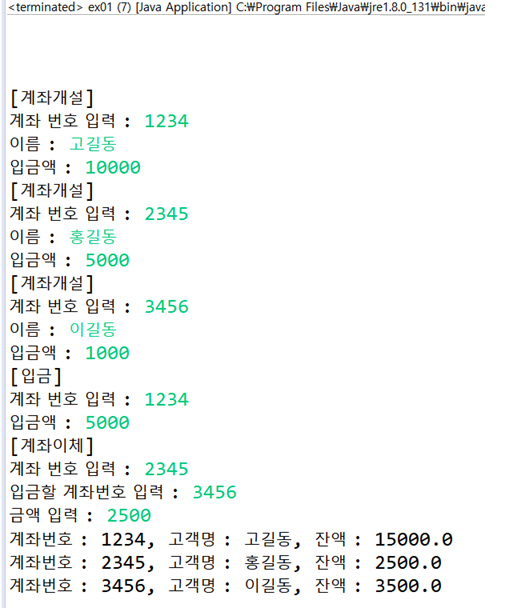
accInfo+=accounts[i].toString()+"\n";

**return** accInfo;

}

}

**■ 결과화면**

****