Day11(8/14)

**package** com.ict.am;

**public** **class** Ex02\_Format {

//Format 클래스 사용

StringBuffer format(String s, **int**... arg) {

StringBuffer sb = **new** StringBuffer();

**int** j = 0;

**for** (**int** i = 0; i < s.length(); i++) {

**char** ch = s.charAt(i);

**if** (ch != '%') {

sb.append(ch);

} **else** {

sb.append(arg[j++]);

}

}

**return** sb;

}

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

// System.out.printf("입력된 값 %d, %d", 500, 700);

Ex02\_Format f = **new** Ex02\_Format();

StringBuffer sbuffer = f.format("입력된 값은 % %", 20, 30);

System.***out***.println(sbuffer);

}

}

\*Import와 패키지

Package:

-서로 관련이 깊은 여러 개의 클래스들을 그룹 지어 놓은 꾸러미

-자바에서 제공되는 많은 클래스들은 아무렇게나 저장해 두었을 경우 찾기 힘들기 때문에

관련된 클래스들을 패키지로 묶어서 관리한다.

String클래스를 사용할 때 마다 일일이 패키지이름까지 기술해야 한다면 클래스 이름이 너무 길어서 무척 불편

🡪import java.lang.\*;과 같이 선언하면 패키지에 포함된 클래스를 모두 사용할 수 있다.

🡪하지만 java.lang 패키지가 자동으로 포함되었기 때문에 import해줄 필요 없다.

**package** com.ict.am;

**import** java.util.ArrayList;

//import구분으로 패키지 지정하기

**public** **class** Ex03\_ArrayList {

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

java.util.ArrayList<String> list = **new** ArrayList();

list.add("하나");

list.add("둘");

list.add("셋");

list.add("넷");

**for**(**int** i=0; i<list.size(); i++) {

System.***out***.println(i+"번째 요소는 "+list.get(i));

}

}

}

\*class 연습

**package** com.ict.bm;

**public** **class** Ex01 {

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

WashingMachine wm1 = **new** WashingMachine();

wm1.setMaker("samsung");

wm1.setWeight(100);

wm1.washing();

wm1.showInfo();

System.***out***.println("::::::");

WashingMachine wm2 = **new** WashingMachine("LG", 150);

wm2.washing();

wm2.showInfo();

}

}

**package** com.ict.bm;

**public** **class** WashingMachine {

**private** String maker;

**private** **int** weight;

// 생성자

**public** WashingMachine() {

}

**public** WashingMachine(String maker, **int** weight) {

**this**.maker = maker;

**this**.weight = weight;

}

// getter

**public** String getMaker() {

**return** maker;

}

**public** **int** getWeight() {

**return** weight;

}

// setter

**public** **void** setMaker(String maker) {

**this**.maker = maker;

}

**public** **void** setWeight(**int** weight) {

**this**.weight = weight;

}

//메소드

**public** **void** washing() {

System.***out***.println("세탁기가 " + getWeight() + "kg의 빨래를 한다.");

}

**public** **void** showInfo() {

System.***out***.println("maker:" + getMaker() + " ,weight:" + getWeight() + "kg");

}

}

**package** com.ict.cm;

**public** **class** SamsungTV {

**private** String tvName;

**private** **int** channel;

//private int volume;

**private** Speaker speaker;

// 생성자

**public** SamsungTV() {}

**public** SamsungTV(String tvName, **int** channel, Speaker speaker) {

**this**.tvName = tvName;

**this**.channel = channel;

//this.volume = volume;

**this**.speaker = speaker;

}

// getter

**public** String getTVName() {

**return** tvName;

}

**public** **int** getChannel() {

**return** channel;

}

// setter

**public** **void** setTVName(String tvName) {

**this**.tvName=tvName;

}

**public** **void** setChnnel(**int** channel) {

**this**.channel = channel;

}

//메소드

**public** **void** soundDown() {

//volume--;

//printVolume();

speaker.volumeUp();

System.***out***.println("볼륨이"+speaker.getVolume()+"입니다.");

}

**public** **void** channelDown() {

channel--;

printChannel();

}

**public** **void** soundUp() {

//volume++;

//printVolume();

speaker.volumeDown();

System.***out***.println("볼륨이"+speaker.getVolume()+"입니다.");

}

**public** **void** channelUp() {

channel++;

printChannel();

}

**public** **void** printChannel() {

System.***out***.println("볼륨이 "+getChannel()+"입니다.");

}

**public** **void** showInfo() {

System.***out***.println(getTVName()+" ,채널:"+getChannel()+" 볼륨"+speaker.getVolume());

}

}

**package** com.ict.cm;

//Speaker를 상속받아 사용

**public** **class** SonySpeaker **extends** Speaker {

**private** **int** volume;

**private** String name = "소니";

**public** SonySpeaker() { }

**public** SonySpeaker(**int** volume) {

**this**.volume = volume;

}

**public** **int** volumeUp() {

volume++;

System.***out***.println(name+"스피커 볼륨을 올린다.");

**return** volume;

}

**public** **int** volumeDown() {

volume--;

System.***out***.println(name+"스피커 볼륨을 내린다.");

**return** volume;

}

//getter

**public** **int** getVolume() {

**return** volume;

}

}

**package** com.ict.cm;

**public** **class** Speaker {

**private** **int** volume;

**public** Speaker() {

**this**.volume = 5;

}

**public** Speaker(**int** volume) {

**this**.volume = volume;

}

**public** **int** volumeUp() {

volume++;

System.***out***.println("기본 스피커 볼륨을 올린다.");

**return** volume;

}

**public** **int** volumeDown() {

volume--;

System.***out***.println("기본 스피커 볼륨을 내린다.");

**return** volume;

}

//getter

**public** **int** getVolume() {

**return** volume;

}

}

**package** com.ict.cm;

**public** **class** TVClient {

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

SamsungTV tv = **new** SamsungTV("벽걸이TV", 7 , **new** SonySpeaker(5));//tvName, channel, volume

tv.soundUp(); //볼륨이 6입니다.

tv.channelUp(); //채널이 8번입니다.

tv.showInfo();//벽걸이 TV-채널:8 볼륨:6

tv.soundDown(); //볼륨이 5입니다.

tv.channelDown(); //채널이 7번입니다.

tv.showInfo();//벽걸이 TV-채널:7 볼륨:5

}

}

\*과제1\*

**package** com.ict.em;

//은행계좌인 Account객체는 잔고(balance)필드를 가지고 있다.

//balance필드는 0~1000000까지만 저장 가능(balance>=0 && balance<=1000000)

//외부에서 balance필드를 변경 할 수없다 ->private

**public** **class** Account {

**private** String ano;

**private** String owner;

**private** **int** balance;

**private** **static** **int** *MIN\_BALANCE* = 0;

**private** **static** **int** *MAX\_BALANCE* = 1000000;

//생성자

**public** Account() {}

**public** Account(String ano, String owner, **int** balance) {

**this**.ano = ano;

**this**.owner = owner;

**this**.balance = balance;

}

//getter

**public** String getAno() {

**return** ano;

}

**public** String getOwner() {

**return** owner;

}

**public** **int** getBalance() {

**return** balance;

}

//setter

**public** **void** setAno(String ano) {

**this**.ano = ano;

}

**public** **void** setOwner(String owner) {

**this**.owner = owner;

}

**public** **void** setBalance(**int** balance) {

**if**(balance>= *MIN\_BALANCE* && balance <= *MAX\_BALANCE*) {

**this**.balance = balance;

}

}

}

**package** com.ict.em;

**public** **class** AccountExample {

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

Account account = **new** Account();

account.setBalance(10000);

System.***out***.println("현재 잔고:" +account.getBalance());

account.setBalance(-100);

System.***out***.println("현재잔고:" +account.getBalance());

account.setBalance(2000000); //100만원 이상이기 때문에 들어가면 안됨

System.***out***.println("현재잔고:" +account.getBalance());

account.setBalance(300000);

System.***out***.println("현재잔고:" +account.getBalance());

}

}

\*과제2\*

**package** com.ict.em;

**import** java.util.Scanner;

**public** **class** BankApplication {

**private** **static** Account[] *accountArr* = **new** Account[100];

**static** Scanner *scan* = **new** Scanner(System.***in***);

**static** **int** *top* = 0;// 현재 위치

**static** **int** *MIN\_BALANCE* = Account.*get\_MIN\_BALANCE*();

**static** **int** *MAX\_BALANCE* = Account.*get\_MAX\_BALANCE*();

**static** {

*makeData*("111-111","park", 10000);

*makeData*("222-222","ryu", 20000);

}

**private** **static** **void** makeData(String ano, String owner, **int** balance) {

*accountArr*[*top*] = **new** Account(ano, owner, balance);

*top*++;

}

// 계좌생성하기

**private** **static** **void** createAccount() {

System.***out***.println("::::계좌생성::::");

System.***out***.print("계좌번호>>");

String ano = *scan*.next();

System.***out***.print("계좌주>>");

String owner = *scan*.next();

System.***out***.print("초기입금액>>");

**int** balance = *scan*.nextInt();

**if** (!(balance >= *MIN\_BALANCE* && balance <= *MAX\_BALANCE*)) {

System.***out***.println(*MIN\_BALANCE* + "~" + *MAX\_BALANCE* + "사이의 금액만 예금 가능합니다.");

System.***out***.println("초기입금액>>");

balance = *scan*.nextInt();

}

*accountArr*[*top*] = **new** Account(ano, owner, balance); // 객체화

System.***out***.println("계좌생성완료!");

*top*++;

}

**public** **static** **void** print(**int** i) {

System.***out***.print(*accountArr*[i].getAno() + "\t");

System.***out***.print(*accountArr*[i].getOwner() + "\t");

System.***out***.print(*accountArr*[i].getBalance() + "\n");

}

// 계좌목록보기

**private** **static** **void** accountList() {

System.***out***.println("::::목록보기::::");

System.***out***.println("계좌번호\t계좌명의\t예금액");

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

*print*(i);

}

}

**private** **static** **void** find() {

System.***out***.println("::::계좌찾기::::");

System.***out***.println("1.계좌 명으로 찾기 2.계좌 번호로 찾기");

String answer = *scan*.next();

Account find\_account = **new** Account();

**if** (answer.equals("1")) {

System.***out***.print("계좌주 입력>>");

String owner = *scan*.next();

find\_account = *findAccount01*(owner);

} **else** **if** (answer.equals("2")) {

System.***out***.print("계좌번호 입력>>");

String ano = *scan*.next();

find\_account = *findAccount*(ano);

} **else** {

System.***out***.println("계좌를 찾을 수 없습니다.");

**return**; // 함수 종료

}

System.***out***.print(find\_account.getAno() + "\t");

System.***out***.print(find\_account.getOwner() + "\t");

System.***out***.print(find\_account.getBalance() + "\n");

}

// 이름으로 객체 찾기

**private** **static** Account findAccount01(String owner) {

Account find\_account = **new** Account();

find\_account.setOwner("-1"); // setOwner가 -1이면 찾지 못한것

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

**if** (owner.equalsIgnoreCase(*accountArr*[i].getOwner())) {

find\_account = *accountArr*[i];

}

}

**return** find\_account;

}

// 계좌번호로 객체 찾기

**private** **static** Account findAccount(String ano) {

Account find\_account = **new** Account();

find\_account.setOwner("-1"); // setOwner가 -1이면 찾지 못한것

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

**if** (ano.equalsIgnoreCase(*accountArr*[i].getAno())) {

find\_account = *accountArr*[i];

}

}

**return** find\_account;

}

// 예금하기 -findAccount(String ano)호출해서 사용하기

**private** **static** **void** deposit() {

System.***out***.println("::::예금::::");

Account find\_account = **new** Account(); // 찾은 객체를 저장할 Account형 객체

System.***out***.print("계좌번호 입력");

String ano = *scan*.next();

find\_account = *findAccount*(ano);

**if** (find\_account.getOwner().equals("-1")) {

System.***out***.println("해당 계좌가 없습니다.");

**return**; // 함수종료

}

System.***out***.print("입금할 금액>>");

**int** deposit = *scan*.nextInt(); // 입금할 금액

**if** ((find\_account.getBalance() + deposit) <= *MAX\_BALANCE*) {

find\_account.setBalance(find\_account.getBalance() + deposit);

System.***out***.println("입금완료!");

} **else** {

System.***out***.println("최대 입금가능 금액:" + *MAX\_BALANCE* + "까지만 입금 할 수 있습니다.");

}

}

// 출금하기

**private** **static** **void** withdraw() {

System.***out***.println("::::출금::::");

Account find\_account = **new** Account(); // 찾은 객체를 저장할 Account형 객체

System.***out***.print("계좌번호 입력");

String ano = *scan*.next();

find\_account = *findAccount*(ano);

**if** (find\_account.getOwner().equals("-1")) {

System.***out***.println("해당 계좌가 없습니다.");

**return**; // 함수종료

}

System.***out***.print("출금할 금액>>");

**int** deposit = *scan*.nextInt(); // 입금할 금액

**if** ((find\_account.getBalance() - deposit) >= *MIN\_BALANCE*) {

find\_account.setBalance(find\_account.getBalance() - deposit);

System.***out***.println("출금완료!");

} **else** {

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

System.***out***.println("현재 출금가능 금액은" + find\_account.getBalance() + "입니다. ");

}

}

**private** **static** **void** delete() {

System.***out***.println("::::계좌 삭제::::");

System.***out***.print("계좌번호 입력");

String ano = *scan*.next();

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

**if** (ano.equalsIgnoreCase(*accountArr*[i].getAno())) {

*print*(i); // 삭제할 정보 출력

**while** (**true**) {

System.***out***.print("정말 계좌를 삭제하시겠습니까?(y/n)");

String answer = *scan*.next();

**if** (answer.equalsIgnoreCase("y")) {

**for** (**int** j = i; j < *top*; j++) {

*accountArr*[j] = *accountArr*[j + 1];

}

System.***out***.println("삭제완료!");

*top*--;

**return**;

} **else** **if** (answer.equalsIgnoreCase("n")) {

System.***out***.println("계좌 삭제기능을 종료하겠습니다.");

**return**;

} **else** {

System.***out***.println("y또는n을 입력해 주세요");

**continue**;

}

}

}

}

System.***out***.println("해당계좌가 존재하지 않습니다.");

}

// 종료

**private** **static** **void** close() {

System.***out***.println("::::종료기능::::");

System.***out***.println("수고하셨습니다");

System.*exit*(0);

}

// Account배열에서 ano와 동일한 Account객체 찾기

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

**while** (**true**) {

System.***out***.println("----------------------------------------------");

System.***out***.println("1.계좌생성 2.계좌목록 3.계좌찾기 4.예금 5.출금 6.삭제 7.종료");

System.***out***.println("----------------------------------------------");

System.***out***.println("선택>>");

String answer = *scan*.next();

**switch** (answer) {

**case** "1":

*createAccount*();

**break**;

**case** "2":

*accountList*();

**break**;

**case** "3":

*find*();

**break**;

**case** "4":

*deposit*();

**break**;

**case** "5":

*withdraw*();

**break**;

**case** "6":

*delete*();

**break**;

**case** "7":

*close*();

**break**;

**default**:

System.***out***.println("1~5 메뉴를 선택해 주세요");

}

}

}

}

\*클래스의 관계

has a: 특정 객체내에서 다른 객체를 가지고 있는 것을 의미한다.

is a: 특정 객체가 다른 객체에게 자신의 능력을 포함시켜주는 상속관계를 의미한다.