**package** main;

**import** manager.GoodsManager;

**public** **class** Main {

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

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

System.***out***.println("[물품정보 관리 프로그램]");

**new** GoodsManager() ;

}

}

**package** manager;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**import** vo.GoodsInfo;

**public** **class** GoodsManager {

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

ArrayList<GoodsInfo> gList = **new** ArrayList<>(); // GoodsInfo 리스트 생성

**public** GoodsManager() {

**boolean** flag = **true**;

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

System.***out***.println("~ 물품정보관리 메뉴 ~");

System.***out***.println("1. 물품 등록");

System.***out***.println("2. 물품 삭제");

System.***out***.println("3. 물품 수정");

System.***out***.println("4. 전체보기");

System.***out***.println("5. 검색하기");

System.***out***.println("6. 물품 출고");

System.***out***.println("메뉴 번호를 선택해주세요.");

**int** selNum = in.nextInt(); // 메뉴번호를 저장할 변수 selNum

in.nextLine();

**if** (selNum == 1) { // 물품 등록

add();

} **else** **if** (selNum == 2) { // 물품 삭제

del();

} **else** **if** (selNum == 3) { // 물품 수정

mod();

} **else** **if** (selNum == 4) { // 전체보기

allList();

} **else** **if** (selNum == 5) { // 검색하기

findGoods();

} **else** **if** (selNum == 6) { // 상품 출고

outGoing();

} **else** {

flag = **false**;

}

}

}

**private** **void** outGoing() {

System.***out***.println("[물품 출고]");

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

allList(); // allList() method 호출

System.***out***.println("출고하시려는 물품을 찾으려면 물품 아이디를 입력해주세요.");

String ogId = in.nextLine(); // 찾으려는 물품 아이디를 저장할 변수 ogId

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

**if** (gList.get(i).getId().equals(ogId)) {

gList.get(i).prt();

System.***out***.println("출고하시는 수량을 입력해주세요.");

**int** outNum = in.nextInt(); // 출고 수량을 저장할 변수 outNum

in.nextLine();

**if** (outNum <= gList.get(i).getCount()) {

// 창고에 있는 수량이 입력한 수보다 많을 경우, 즉 있는 수량 내에서 수량을 입력한 경우

gList.get(i).setCount(gList.get(i).getCount() - outNum);

// gList.get(i)를 참조해 getCount()로 저장된 수량을 가져온 뒤 outNum을 빼줌

// 위 값을 gList.get(i)로 해당 값을 참조해 setCount()

System.***out***.println("출고 되었습니다.");

gList.get(i).prt();

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

**break**; // 리턴값

} **else** {

System.***out***.println("창고에 있는 물품의 수량보다 입력하신 수량이 더 많습니다.");

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

}

} **else** {

System.***out***.println("해당하는 물품이 없습니다.");

}

}

}

**public** **void** findGoods() {

System.***out***.println("[물품 찾기]");

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

System.***out***.println("찾으시려는 물품명을 입력해주세요.");

String fdName = in.nextLine(); // 검색명 저장하는 변수 fdName

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

**if** (gList.get(i).getName().contains(fdName)) {

gList.get(i).prt();

// 객체의 주소를 가져와서 그 주소의 값을 참조해서 프린트한다.

} **else** {

System.***out***.println("찾으시는 물품이 없습니다.");

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

}

}

}

**public** **void** add() { // 물품을 등록하는 method 정의

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

System.***out***.println("[물품 등록]");

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

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

System.***out***.println("아이디를 입력해주세요.");

System.***out***.println("아이디는 중복될 수 없습니다.");

String id = in.nextLine();

**if** (chkId(id) == **false**) { // chkId method를 사용하여 아이디가 중복이 아닌 경우

System.***out***.println("물품명을 입력해주세요.");

String name = in.nextLine();

System.***out***.println("수량을 입력해주세요.");

System.***out***.println("(숫자만 입력)");

**int** count = in.nextInt();

in.nextLine();

GoodsInfo temp = **new** GoodsInfo();

temp.setId(id);

temp.setName(name);

temp.setCount(count);

gList.add(temp); // gList에 id, name, count 추가

**return**;

}

}

}

**public** **boolean** chkId(String ids) { // 중복 아이디 체크하는 method 정의

// 아이디 체크할 때 넣으면 리스트 사이즈가 가변적이라 0으로 시작하기에 계속 아래 코드가 실행될 수 없음.

// 따라서 아이디 중복 체크를 위한 method 정의

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

**if** (gList.get(i).getId().equals(ids) == **true**) {

// 리스트에 저장한 id를 get(리스트 인덱스번호).getId()로 불러와

// 위에서 입력받은 아이디가 저장된 ids로 값을 받아와서

// 만약 리스트에 저장된 값이 동일할 경우 아이디 중복 알림 띄우기

System.***out***.println("\* 아이디가 중복되었습니다. \*");

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

**return** **true**;

}

}

**return** **false**; // 아이디가 중복이 아님

}

**public** **void** allList() { // 등록된 물품을 볼 수 있는 method 정의

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

System.***out***.println("[등록된 물품 리스트]");

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

System.***out***.println("물품 번호 : " + i);

gList.get(i).prt(); // i번째 인덱스의 주소값 출력

}

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

}

**public** **void** del() { // 물품을 삭제하는 method 정의

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

System.***out***.println("[물품 삭제]");

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

allList(); // allList() method 호출

System.***out***.println("삭제하실 물품 번호를 입력해주세요.");

**int** delNum = in.nextInt(); // 삭제할 물품 번호를 저장하는 변수 delNum

in.nextLine();

gList.remove(delNum); // 인덱스 번호가 delNum인 인덱스 값 제거

}

**public** **void** mod() { // 물품을 수정하는 method 정의

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

System.***out***.println("[물품 수정]");

allList(); // allList() method 호출

System.***out***.println("수정하실 물품 번호를 입력해주세요.");

System.***out***.println("수정은 물품명과 수량만 수정이 가능합니다.(아이디는 불가)");

**int** modNum = in.nextInt(); // 수정할 물품 번호를 저장하는 변수 modNum

in.nextLine();

System.***out***.println("수정하실 물품명을 입력해주세요.");

System.***out***.println("기존 물품명 : " + gList.get(modNum).getName());

String name = in.nextLine(); // 수정할 물품명 입력

System.***out***.println("수정하실 수량을 입력해주세요.");

System.***out***.println("기존 수량 : " + gList.get(modNum).getCount());

**int** count = in.nextInt(); // 수정할 수량 입력

in.nextLine();

gList.get(modNum).setName(name); // 인덱스 번호가 modNum인 name 값을 수정

gList.get(modNum).setCount(count); // 인덱스 번호가 modNum인 count 값을 수정

}

}

**package** vo;

**import** manager.GoodsManager;

**public** **class** GoodsInfo {

GoodsManager gm = **null** ;

**private** String id ; // 물품 아이디. 중복된 아이디는 없음.

**private** String name ; // 물품명

**private** String count ; // 수량

**public** String getId() { // id를 넘김

**return** id;

}

**public** **void** setId(String id) { // id의 주소값을 저장

**this**.id = id;

}

**public** String getName() { // 물품명 주소값을 넘김

**return** name;

}

**public** **void** setName(String name) { // name의 주소값을 저장

**this**.name = name;

}

**public** String getCount() { // 수량 주소값을 넘김

**return** count;

}

**public** **void** setCount(String count) { // count의 주소값을 저장...?

**this**.count = count;

}

// public boolean chkId(String ids) { // 중복 아이디 체크하는 method 정의

// for (int i = 0 ; i < gm.gList.size() ; i++) {

// if (gm.gList[i].id.equals(ids)) {

//

// }

// }

// }

**public** **void** prt() { // 저장된 리스트 값을 출력하는 method 정의

System.***out***.println("아이디 : " + id);

System.***out***.println("물품명 : " + name);

System.***out***.println("수량 : " + count);

}

}