

1. 구현 클래스 다이어그램 (Class Diagram)

Run
+ <u>main(args:String[]) : void</u>

MemberMenu
- sc:Scanner = Menu.sc - mc:MemberController = new MemberController() - fm : FarmMenu = new FarmMenu() + <u>loginMember:Member</u>
+ login():Member + enroll():void + customerMenu():void + adminMenu():void + checkMyInfo():void + updateMyInfo():void + deleteMyInfo():void + goVegetable():void + logout():void

Member
- id:String - pwd:String - name:String - isManager:boolean
+ Member() + Member(id:String, pwd:String, name:String, isManager:boolean) + setter() / getter() + toString() : String

Menu
+ <u>sc:Scanner = new Scanner(System.in)</u> - mm:MemberMenu = new MemberMenu() + mainMenu() : void

MemberController
- md:MemberDAO = new MemberDAO() + <u>loginMember:Member</u>
+ login(id:String, pwd:String):Member + enroll(m:Member):void + updateMember(updateMember:Member):void + deleteMember(member:Member):void + checkId(id:String):boolean

MemberDAO
+ selectAll():ArrayList<Member> + checkId(id:String):boolean + enroll(m:Member):void + updateMember(updateMember:Member):void + insertMember(list:ArrayList<Member>):void + deleteMember(member:Member):void

Purchase
- buyer:String - date:String - product:String - price:int - num:int - total:int
+ Purchase() + Purchase(buyer:String, date:String, product:String, price:int, num:int, total:int) + setter() / getter() + toString() : String

FarmMenu
- sc:Scanner = Menu.sc - fc : FarmController = new FarmController()
+ customerMenu() : void + addNewKind() : void + updateKind():void + removeKind() : void + printFarm() : void + buyFarm() : void + refundFarm() : void + printBuyFarm() : void + printSales():void

Farm
- kind:String - price:int - num:int
+ Farm() + Farm(kind:String, price:int, num:int) + setter() / getter() + toString() : String

FarmController
- fd:FarmDAO = new FarmDAO() + <u>farm:HashMap<String, Farm></u>
+ FarmController() + printFarm() : HashMap<String, Farm> + addNewKind(name:String, f:Farm):Farm + insertNewKind():void + checkFarm(name:String):boolean + changeName(originName:String, newName:String):void + changeNumber(menuNum:int, name:String, newNum:int):void + removeFarm(name:String) : void + buyFarm(name:String, price:int, num:int, total:int) : void + printBuyFarm() : ArrayList<Purchase> + refund(list:ArrayList<Purchase>) : void + selectAllBuying() : ArrayList<Purchase>

FarmDAO
+ insertFarm(farm:HashMap<String, Farm>):void + selectAll():HashMap<String, Farm> + insertBuyFarm(pList:ArrayList<Purchase>):void + selectAllBuying():ArrayList<Purchase>

2. 구현 클래스 설명

Package명	Class명	Method	설명
com.kh.overall.run	Run	<u>+main(args:String[]) : void</u>	Menu 객체를 생성 후 mainMenu() 실행
com.kh.overall.view	Menu	+mainMenu():void	첫 시작 출력 메소드
com.kh.overall.member.view	MemberMenu	+login():Member	일반 회원 및 관리자 로그인 뷰 메소드
		+enroll():void	일반 회원 및 관리자 회원가입 뷰 메소드
		+customerMenu():void	일반 회원 메뉴 뷰 메소드
		+ adminMenu():void	관리자 메뉴 뷰 메소드
		+ checkMyInfo():void	일반 회원 정보 확인 뷰 메소드
		+ updateMyInfo():void	일반 회원 정보 수정 뷰 메소드
		+ deleteMyInfo():void	일반 회원 탈퇴 뷰 메소드
		+ goVegetable():void	농산물 뷰 이동 메소드
		+ logout():void	로그아웃 뷰 메소드
com.kh.overall.view	FarmMenu	+ customerMenu() : void	일반 회원 농산물 뷰 메소드
		+ adminMenu() : void	관리자 농산물 뷰 메소드
		+addNewKind() : void	새로운 농산물 추가 뷰 메소드
		+ updateKind():void	농산물 수정 뷰 메소드
		+removeKind() : void	농산물 삭제 뷰 메소드
		+printFarm():void	농산물 출력 뷰 메소드
		+buyFarm():void	농산물 구매 뷰 메소드
		+ refundFarm() : void	농산물 환불 뷰 메소드
		+printBuyFarm():void	구매 목록 출력 메소드
		+ printSales():void	총 매출 출력 메소드
com.kh.overall.member.controller	MemberController	+ login(id:String, pwd:String):Member	전달받은 아이디와 비밀번호가 존재하는지 확인하는 메소드
		+ enroll(m:Member):void	전달받은 데이터를 등록하고자 DAO에 전달하는 메소드
		+updateMember(updateM ember:Member):void	전달받은 데이터로 수정하고자 DAO에 전달하는 메소드
		+deleteMember(member: Member):void	전달받은 데이터를 삭제하고자 DAO에 전달하는 메소드
		+checkId(id:String):boolean	전달받은 아이디가 존재하는지 확인하기 위해 DAO에 전달하는 메소드

com.kh.overall.farm .controller	FarmController	+FarmController()	기본 생성자
		+printFarm() : HashMap<String, Farm>	클래스변수 farm 반환 메소드
		+addNewKind(name:String, f:Farm):Farm	전달받은 데이터를 farm에 등록 한 결과를 반환하는 메소드
		+ insertNewKind():void	farm을 저장하고자 DAO에 전달 하는 메소드
		+checkFarm(name:String):b oolean	전달받은 데이터의 존재 여부를 반환하는 메소드
		+changeName(originName :String, newName:String):void	기존 농산물 명을 다른 농산물 명으로 바꾸기 위해 DAO에 전달 하는 메소드
		+changeNumber(menuNu m:int, name:String, newNum:int):void	기존 농산물 가격/수를 다른 값 으로 바꾸기 위해 DAO에 전달하 는 메소드
		+removeFarm(name:String) : void	등록된 농산물이 제거된 farm을 DAO에 전달하는 메소드
		+buyFarm(name:String, price:int, num:int, total:int) : void	농산물을 사기 위해 DAO에 전달 하는 메소드
		+ printBuyFarm() : ArrayList<Purchase>	자신이 구매한 농산물을 DAO로 부터 반환하는 메소드
		+refund(list:ArrayList<Purc hase>) : void	구매했던 농산물을 환불하기 위 해 DAO에 전달하는 메소드
		+ selectAllBuying() : ArrayList<Purchase>	구매한 농산물을 DAO로부터 반 환하는 메소드
com.kh.overall.me mber.model.dao	MemberDAO	+selectAll():ArrayList<Mem ber>	파일에 저장된 데이터를 불러와 ArrayList에 담아 반환
		+checkId(id:String):boolean	파일에 저장된 아이디 중 전달받 은 아이디와의 일치 여부 반환
		+enroll(m:Member):void	전달받은 데이터를 파일에 저장
		+updateMember(updateM ember:Member):void	전달받은 데이터로 파일 수정
		+insertMember(list:ArrayLis t<Member>):void	전달받은 데이터를 파일에 덮어 써 저장
		+deleteMember(member: Member):void	전달받은 데이터를 삭제하기 위 해 파일에 덮어써 저장
com.kh.overall.farm .model.dao	FarmDAO	+insertFarm(farm:HashMap <String, Farm>):void	전달받은 데이터를 파일에 덮어 써 저장

		+selectAll():HashMap<String, Farm>	파일에 저장된 데이터를 불러와 HashMap에 담아 반환
		+insertBuyFarm(pList:ArrayList<Purchase>):void	전달받은 데이터를 파일에 덮어써 저장
		+selectAllBuying():ArrayList<Purchase>	파일에 저장된 데이터를 불러와 ArrayList에 담아 반환
com.kh.overall.member.model.vo	Member	+Member()	기본 생성자
		+Member(id:String, pwd:String, name:String, isManager:boolean)	매개변수가 있는 생성자
		+ setter() / getter()	필드 값을 수정/반환하는 메소드
		+ toString() : String	객체 정보 리턴 메소드 "id/pwd/name/isManager" 형식
com.kh.overall.farm.model.vo	Farm	+ Farm()	기본 생성자
		+ Farm(kind:String, price:int, num:int)	매개변수가 있는 생성자
		+ setter() / getter()	필드 값을 수정/반환하는 메소드
		+ toString() : String	객체 정보 리턴 메소드 "(kind) price원 : num개" 형식
	Purchase	+ Purchase()	기본 생성자
		+ Purchase(buyer:String, date:String, product:String, price:int, num:int, total:int)	매개변수가 있는 생성자
		+ setter() / getter()	필드 값을 수정/반환하는 메소드
		+ toString() : String	객체 정보 리턴 메소드 "buyer/date/product price원(개당) num개, 총 total원" 형식

3. class 구조

```
public class Run{

    public static void main(String[] args){

        new Menu().mainMenu();

    }

}

public class Menu{

    // Scanner 객체 생성
    // MemberMenu 객체 생성(mm)
    public void mainMenu(){

        System.out.println("===== 환영합니다. =====");
        ***** 메인 메뉴 *****

        1. 로그인                // mm.login() 실행
        2. 회원가입              // mm.enroll() 실행
        9. 종료                  // "프로그램 종료." 출력 후 main()으로 리턴
        메뉴 번호 선택 : >> 입력 받음 // 메뉴 화면 반복 실행 처리
        // 잘 못 입력 하였을 경우 "잘못 입력하셨습니다. 다시 입력해주세요" 출력 후 반복

    }

}
```

```
public class MemberMenu{

    // Menu에 생성했던 Scanner객체를 sc필드에 저장
    // MemberController 객체 생성(mc)
    // FarmMenu 객체 생성(fm)
    // Member타입의 loginMember필드 생성

    public Member login(){
        아이디 :
        비밀번호 :
        // 아이디와 비밀번호를 사용자에게 입력 받아 존재하는 회원인지 확인
        // 계정이 없으면 "존재하지 않는 회원입니다."
        // 계정이 있으면 "OOO님, 환영합니다." 출력
        // 해당 회원이 일반 회원인지 관리자인지 확인 후
        // 일반 회원이면 customerMenu()로 이동, 관리자면 adminMenu()로 이동
    }

    public void enroll(){
        // 아이디, 비밀번호, 이름, 관리자 여부를 사용자에게 입력 받음
        // 입력 받은 아이디가 존재하는지 확인 후 기존에 저장되지 않은(중복되지 않은)
        // 아이디가 입력될 때까지 반복
        // 이때 아이디가 중복 됐으면 "중복된 아이디입니다. 다시 입력해주세요"
        // 중복되지 않은 아이디면 "사용가능한 아이디입니다." 출력
        // 비밀번호, 이름까지 모두 입력 받고 관리자인지 사용자에게 물어봄
        // 관리자가 맞다고 하면 관리자 코드를 입력 받아 일치했을 때 관리자로 회원가입,
        // 관리자 코드가 일치하지 않으면 관리자 코드가 아니라는 안내와 함께
        // 다시 한 번 더 관리자인지 물어봄
        // 관리자가 아니라고 답하면 일반 회원으로 회원가입
        // 회원가입에 성공하면 "회원가입이 성공하였습니다" 출력
    }
}
```

```

public void customerMenu(){
    ===== 일반 회원 메뉴 =====
    1. 내 정보 보기           // checkMyInfo() 실행
    2. 내 정보 수정           // updateMyInfo() 실행
    3. 탈퇴하기               // deleteMyInfo() 실행
    4. 농산물 메뉴로 가기     // goVegetable() 실행
    5. 로그아웃               // logout(); 실행 후 return
    메뉴 번호 입력 : >> 입력 받음 // 메뉴 화면 반복 실행 처리
    // 잘 못 입력 하였을 경우 "잘못 입력하셨습니다. 다시 입력해주세요" 출력 후 반복
    // loginMember가 null이면 return
}

```

```

public void adminMenu(){
    ===== 직원 메뉴 =====
    1. 새 농산물 추가         // fm.addNewKind () 실행
    2. 농산물 수정           // fm.updatekind() 실행
    3. 농산물 삭제           // fm.removeKind() 실행
    4. 농산물 목록           // fm.printFarm() 실행
    5. 매출 보기             // fm.printSales(); 실행
    6. 로그아웃              // logout() 실행
    메뉴 번호 입력 : >> 입력 받음 // 메뉴 화면 반복 실행 처리
    // 잘 못 입력 하였을 경우 "잘못 입력하셨습니다. 다시 입력해주세요" 출력 후 반복
    // loginMember가 null이면 return
}

```



```
public void checkMyInfo(){
    // loginMember에 저장된 아이디, 비밀번호, 이름 출력
    // 이때 비밀번호는 앞 두 자리 뒤에 ***** 붙여서 출력
}

public void updateMyInfo(){
    // 현재 내 정보를 먼저 출력해서 보여준 후 비밀번호, 이름 중 변경할 수 있도록 함
    // 더 바꿀 건지에 대한 대답으로 n이 나올 때까지 반복
}

public void deleteMyInfo(){
    정말 탈퇴하시겠습니까?(Y/N) :
    // 탈퇴 여부에 대한 대답을 입력 받아 탈퇴하겠다고 하면 탈퇴 후 로그아웃, 메인
    // 탈퇴하지 않겠다고 하면 그대로 메인으로 넘어감
}

public void goVegetable(){
    // fm.customerMenu()로 이동
}

public void logout(){
    로그아웃합니다.
    // loginMember를 null로 만들
}
}
```

```

public class FarmMenu {

    // Menu에 생성했던 Scanner객체를 sc필드에 저장
    // FarmController 객체 생성

    public void customerMenu() {

        현재 KH마트 농산물 수량

        printFarm();

        ***** 고객 메뉴 *****

        1. 농산물 구입하기                // buyFarm() 실행
        2. 농산물 환불하기                // refundFarm() 실행
        3. 구입한 농산물 보기              // printBuyFarm() 실행
        9. 메인으로 돌아가기              // return

        메뉴 번호 입력 : >> 입력 받음    // 메뉴 화면 반복 실행 처리

        // 잘못 입력 하였을 경우 "잘못 입력하였습니다. 다시 입력해주세요" 출력 후 반복
    }

    public void addNewKind() {

        // "1. 과일 / 2. 채소 / 3. 견과"를 통해 번호로 종류를 받음
        // 번호가 벗어나면 "잘못된 번호입니다." 출력 후 제대로 받아올 때까지 반복
        // 농산물의 이름, 수량, 가격도 함께 입력 받음
        // 농산물의 종류(문자열로 변환), 수량, 가격 데이터를 Farm객체 f 생성 후 저장
        // 이전에 저장 되어있던 농산물 데이터를 HashMap before에 저장
        // 그리고 방금 사용자에게 받은 데이터가 before에 있는지 확인
        // 없다면 "새 농산물이 추가되었습니다." 출력
        // 있다면 새로 넣을 데이터로 변경할 것인지 물어보고
        // 바꾸겠다고 하면 새 데이터로, 유지하겠다고 하면 이전 데이터로 저장
    }
}

```

```
public void updateKind() {  
    //현재 저장 되어있는 농산물을 먼저 출력하고, 수정할 부분(농산물 명, 수량, 가격)을  
    //번호로 입력 받음  
    //수정하고자 하는 농산물 명을 입력하고 해당 농산물이 없으면  
    //"해당 농산물을 존재하지 않습니다" 출력 있으면 뭘로 변경할지 입력 받아 수정  
}  
  
public void removeKine(){  
    // 현재 저장 되어있는 농산물을 먼저 출력하고, 삭제할 농산물 명을 입력받음  
    // 해당 농산물이 없으면 "해당 농산물을 존재하지 않습니다" 출력  
    // 있으면 "농산물 삭제에 성공하였습니다." 출력  
    // 더 진행할 건지 물어보고 더 진행하겠다고 하면 반복 아니면 반복 빠져나감  
}  
  
public void printFarm(){  
    // fc로부터 전달받은 HashMap을 통해서 비어 있으면 "등록된 농산물이 없습니다"  
    // 데이터가 있으면 모두 출력  
}  
  
public void buyFarm(){  
    // 구매할 농산물 명을 입력 후 없으면 "해당 농산물은 존재하지 않습니다" 출력,  
    // 있으면 몇 개를 구매할지 수량 입력  
    // 저장 되어있는 수량보다 많이 쓰면 "준비된 수량이 부족합니다" 출력  
    // 저장 되어있는 수량보다 작거나 같으면 구매하겠냐고 물어본 후  
    // 구매하겠다고 했을 때 구매 정보를 fc로 넘김  
    // 더 진행할 건지 물어보고 더 진행하겠다고 하면 반복 아니면 반복 빠져나감  
}
```

```

public void refundFarm(){
    // 구매한 농산물 정보를 번호와 함께 출력
    // 환불할 농산물을 번호로 입력 받고 해당 농산물을 구입목록에서 삭제
}

public void printBuyFarm(){
    // 구매한 농산물 정보를 fc로부터 가져오는데 비어 있으면 "구매이력이 없습니다"
    // 있다면 구매 정보 출력
}

public void printSales(){
    // 저장된 모든 구매 이력들을 모두 출력하고 그에 따른 총 매출까지 출력
}
}

```

```

public class MemberController{
    // MemberDAO 객체 (md)
    // Member loginMember 변수
    public Member login(String id, String pwd){
        // DAO에서 반환 받은 정보들을 가지고 입력받은 id와 pwd가 있는지 확인 후
        // 일치하는 정보가 있다면 해당 Member 반환, 없다면 null 반환
    }

    public void enroll(Member m){
        // 전달 받은 데이터를 md.enroll()에 전달
    }

    public void updateMember(Member updateMember){
        // 전달 받은 데이터를 md.updateMember()에 전달
    }

    public void deleteMember(Member member){
        // 전달 받은 데이터를 md.deleteMember()에 전달
    }

    public boolean checkId(String id){
        // 전달 받은 데이터를 md.checkId()에 전달 후 해당 메소드의 반환 값 리턴
    }
}

```

```
public class FarmController{

    // FarmDAO객체 생성(fd)
    // HashMap<String, Farm> farm 변수
    public FarmController(){

        // fd.selectAll()을 통해 반환받은 결과 값을 farm에 저장
    }

    public HashMap<String, Farm> printFarm(){

        // farm 반환
    }

    public Farm addNewKind(String name, Farm f){

        // farm에 전달받은 데이터 저장 후 반환 받은 리턴 값 반환
    }

    public void insertNewKind(){

        // fd.insertFarm()에 farm 전달
    }

    public boolean checkFarm(String name){

        // farm에 전달받은 데이터가 키로 존재하는지 여부 반환
    }

    public void changeName(String originName, String newName){

        // farm에 originName에 대한 value값 추출 후, 추출된 value와 newName으로 저장
        // 기존 originName은 삭제 후 변경된 farm을 fd.insertFarm에 전달하여 저장
    }

    public void changeNumber(int menuNum, String name, int newNum){

        // menuNum으로 수량 변경인지 가격 변경인지 구분 후 해당 값을
        // fd.insertFarm에 전달하여 저장
    }

    public void removeFarm(String name){

        // 전달받은 name으로 farm에서 삭제, 그 farm을 다시 fd.insertFarm에 전달 저장
    }

}
```

```

public void buyFarm(String name, int price, int num, int total){
    // 오늘 날짜를 yyyy-MM-dd형식으로 만들어서
    // 구매자 아이디와 날짜, 농산물 명, 가격, 수량, 총 금액을 Purchase 객체에 저장
    // p를 fd.selectAllBuying()으로 전달받은 ArrayList pList에 저장하여 pList를 다시
    // fd.insertBuyFarm()에 전달하여 저장
}

public ArrayList<Purchase> printBuyFarm(){
    // 로그인한 사람의 구매 정보만 가져올 수 있도록 fd.selectAllBuying()에서
    // 반환 받은 ArrayList로 id와 비교함
    // 해당 정보들을 새 ArrayList에 담아 그 ArrayList 반환
}

public void refund(ArrayList<Purchase> list){
    // 전달받은 list를 fd.insertBuyFarm()에 전달
}

public ArrayList<Purchase> selectAllBuying(){
    // fd.selectAllBuying()의 반환 값 다시 리턴
}

}

public class MemberDAO{
    public ArrayList<Member> selectAll(){
        // member.txt를 읽어와 ArrayList<Member>에 저장하고 반환
    }

    public boolean checkId(String id){
        // selectAll()로 받아온 ArrayList에서 전달 받은 아이디가 있는지 여부를 반환
    }

    public void enroll(Member m){
        // 넘겨받은 Member m의 toString() 값과 줄바꿈을 member.txt에 씀
    }
}

```

```

public void updateMember(Member updateMember){
    // selectAll()로 받아온 ArrayList list에서 전달받은 아이디가 있는지 확인하고
    // 아이디가 있다고 했을 때
    // 전달받은 이름이 null이 아니면 list 중 해당 객체 이름을, 전달받은 이름으로 변경
    // 전달받은 pwd가 null이 아니면 list 중 해당 객체 pwd를 전달 받은 pwd로 변경
    // 그 후 수정된 list를 insertMembers()메소드의 인자로 넘김
}

public void insertMembers(ArrayList<Member> list){
    // 전달 받은 list의 객체의 toString()값과 줄바꿈을 member.txt에 저장
}

public void deleteMember(Member member){
    // selectAll()로 받아온 ArrayList list에서 전달받은 아이디가 있는 지 확인하고
    // 있으면 해당 list의 해당 객체를 지움
    // 그 후 수정된 list를 insertMembers()메소드의 인자로 넘김
}
}

public class FarmDAO{
    public void insertFarm(HashMap<String, Farm> farm){
        // 전달 받은 farm을 farm.txt에 쓰면서 저장
        // 이 때, key/kind/num/price 형식으로 저장
    }

    public HashMap<String, Farm> selectAll(){
        // 새로운 HashMap fam 생성 후 farm.txt에 있는 내용을 읽어 fam에 저장 및 리턴
    }

    public void insertBuyFarm(ArrayList<Purchase> pList){
        // buy.txt에 전달받은 pList를 저장
        // buyer/date/product/price/num/total 형식으로 저장
    }

    public ArrayList<Purchase> selectAllBuying(){
        // 새로운 ArrayList list를 생성 후 buy.txt에 있는 내용을 읽어 list에 저장 및 리턴
    }
}

```

===== 환영합니다. =====

=====***** 메인 메뉴 *****=====

1. 로그인
2. 회원가입
9. 종료

메뉴 번호 입력 : 1

아이디 : admin

비밀번호 : ad

존재하지 않는 회원입니다.

=====***** 메인 메뉴 *****=====

1. 로그인
2. 회원가입
9. 종료

메뉴 번호 입력 : 2

아이디 : admin

사용가능한 아이디입니다.

비밀번호 : ad

이름 : 나관리

당신은 관리자입니까?(Y/N) : k

잘못 입력하셨습니다. 다시 입력해주세요.

당신은 관리자입니까?(Y/N) : Y

관리자 코드를 입력하세요 : rosemary4110ZZang

관리자임이 확인되었습니다. 관리자로 회원가입이 진행됩니다.

회원가입이 성공하였습니다.

=====***** 메인 메뉴 *****=====

1. 로그인
2. 회원가입
9. 종료

메뉴 번호 입력 : 2

아이디 : admin

중복된 아이디입니다. 다시 입력해주세요.

아이디 : rose

사용가능한 아이디입니다.

비밀번호 : ro

이름 : 고객임

당신은 관리자입니까?(Y/N) : y

관리자 코드를 입력하세요 : 몰라

관리자 코드가 아닙니다.

당신은 관리자입니까?(Y/N) : n

일반 회원으로 회원가입이 진행됩니다.

회원가입이 성공하였습니다.

=====***** 메인 메뉴 *****=====

1. 로그인

2. 회원가입

9. 종료

메뉴 번호 입력 : 1

아이디 : admin

비밀번호 : ad

나관리님, 환영합니다.

=====***** 직원 메뉴 *****=====

1. 새 농산물 추가

2. 농산물 수정

3. 농산물 삭제

4. 농산물 목록

5. 매출 보기

6. 로그아웃

메뉴 번호 입력 : 4

등록된 농산물이 없습니다.

=====***** 직원 메뉴 *****=====

1. 새 농산물 추가

2. 농산물 수정

3. 농산물 삭제

4. 농산물 목록

5. 매출 보기

6. 로그아웃

메뉴 번호 입력 : 5

등록된 농산물이 없습니다.

아직 매출이 발생하지 않았습니다.

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 1

1. 과일 / 2. 채소 / 3. 견과

추가할 종류 번호 : 1

이름 : 사과

수량 : 1000

가격 : 1500

새 농산물이 추가되었습니다.

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 4

사과 (과일) 1500원 : 1000개

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 1

1. 과일 / 2. 채소 / 3. 견과

추가할 종류 번호 : 1

이름 : 복숭아

수량 : 500

가격 : 2000

새 농산물이 추가되었습니다.

===== 직원 메뉴 =====

1. 새 농산물 추가

2. 농산물 수정

3. 농산물 삭제

4. 농산물 목록

5. 매출 보기

6. 로그아웃

메뉴 번호 입력 : 1

1. 과일 / 2. 채소 / 3. 견과

추가할 종류 번호 : 2

이름 : 오이

수량 : 100

가격 : 1500

새 농산물이 추가되었습니다.

===== 직원 메뉴 =====

1. 새 농산물 추가

2. 농산물 수정

3. 농산물 삭제

4. 농산물 목록

5. 매출 보기

6. 로그아웃

메뉴 번호 입력 : 1

1. 과일 / 2. 채소 / 3. 견과

추가할 종류 번호 : 3

이름 : 땅콩

수량 : 9999

가격 : 2

새 농산물이 추가되었습니다.

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 4

복숭아 (과일) 2000원 : 500개
땅콩 (견과) 2원 : 9999개
사과 (과일) 1500원 : 1000개
오이 (채소) 1500원 : 100개

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 2

복숭아 (과일) 2000원 : 500개
땅콩 (견과) 2원 : 9999개
사과 (과일) 1500원 : 1000개
오이 (채소) 1500원 : 100개

무엇을 수정하시겠습니까?

1. 농산물 명
2. 수량
3. 가격
9. 그만하기

메뉴 번호 입력 : 1

수정할 농산물 명을 입력하세요 : 배추

해당 농산물은 존재하지 않습니다.

복숭아 (과일) 2000원 : 500개

땅콩 (견과) 2원 : 9999개

사과 (과일) 1500원 : 1000개

오이 (채소) 1500원 : 100개

무엇을 수정하시겠습니까?

1. 농산물 명
2. 수량
3. 가격
9. 그만하기

메뉴 번호 입력 : 1

수정할 농산물 명을 입력하세요 : 사과

변경할 값 : 청사과

복숭아 (과일) 2000원 : 500개

땅콩 (견과) 2원 : 9999개

청사과 (과일) 1500원 : 1000개

오이 (채소) 1500원 : 100개

무엇을 수정하시겠습니까?

1. 농산물 명
2. 수량
3. 가격
9. 그만하기

메뉴 번호 입력 : 2

수정할 농산물 명을 입력하세요 : 복숭아

변경할 값 : 150

복숭아 (과일) 2000원 : 150개

땅콩 (견과) 2원 : 9999개

청사과 (과일) 1500원 : 1000개

오이 (채소) 1500원 : 100개

무엇을 수정하시겠습니까?

1. 농산물 명
2. 수량
3. 가격
9. 그만하기

메뉴 번호 입력 : 3

수정할 농산물 명을 입력하세요 : **오이**

변경할 값 : **1000**

복숭아 (과일) 2000원 : 150개

땅콩 (견과) 2원 : 9999개

청사과 (과일) 1500원 : 1000개

오이 (채소) 1000원 : 100개

무엇을 수정하시겠습니까?

1. 농산물 명
2. 수량
3. 가격
9. 그만하기

메뉴 번호 입력 : **0**

잘못 입력하셨습니다. 다시 입력해주세요.

복숭아 (과일) 2000원 : 150개

땅콩 (견과) 2원 : 9999개

청사과 (과일) 1500원 : 1000개

오이 (채소) 1000원 : 100개

무엇을 수정하시겠습니까?

1. 농산물 명
2. 수량
3. 가격
9. 그만하기

메뉴 번호 입력 : **9**

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : **4**

복숭아 (과일) 2000원 : 150개

땅콩 (견과) 2원 : 9999개

청사과 (과일) 1500원 : 1000개

오이 (채소) 1000원 : 100개

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 3

복숭아 (과일) 2000원 : 150개

땅콩 (견과) 2원 : 9999개

청사과 (과일) 1500원 : 1000개

오이 (채소) 1000원 : 100개

삭제할 농산물 명을 입력하세요 : 호두

해당 농산물은 존재하지 않습니다.

더 진행하시겠습니까?(Y/N) : k

잘못 입력하셨습니다. 다시 입력하세요.

더 진행하시겠습니까?(Y/N) : n

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 4

복숭아 (과일) 2000원 : 150개

땅콩 (견과) 2원 : 9999개

청사과 (과일) 1500원 : 1000개

오이 (채소) 1000원 : 100개

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 3

복숭아 (과일) 2000원 : 150개
땅콩 (견과) 2원 : 9999개
청사과 (과일) 1500원 : 1000개
오이 (채소) 1000원 : 100개

삭제할 농산물 명을 입력하세요 : 땅콩
농산물 삭제에 성공하였습니다.
더 진행하시겠습니까?(Y/N) : y

복숭아 (과일) 2000원 : 150개
청사과 (과일) 1500원 : 1000개
오이 (채소) 1000원 : 100개

삭제할 농산물 명을 입력하세요 : 청사과
농산물 삭제에 성공하였습니다.
더 진행하시겠습니까?(Y/N) : n

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 4

복숭아 (과일) 2000원 : 150개
오이 (채소) 1000원 : 100개

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 1

1. 과일 / 2. 채소 / 3. 견과

추가할 종류 번호 : 3

이름 : 호두

수량 : 3000

가격 : 5

새 농산물이 추가되었습니다.

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 4

복숭아 (과일) 2000원 : 150개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 100개

===== 직원 메뉴 =====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 6

로그아웃합니다.

=====***** 메인 메뉴 *****=====

1. 로그인
2. 회원가입
9. 종료

메뉴 번호 입력 : 1

아이디 : rose

비밀번호 : ro

고객임님, 환영합니다.

=====***** 일반 회원 메뉴 *****=====

1. 내 정보 보기
2. 내 정보 수정
3. 탈퇴하기
4. 농산물 메뉴로 가기
5. 로그아웃

메뉴 번호 입력 : 1

ID : rose
PWD : ro*****
NAME : 고객임

=====***** 일반 회원 메뉴 *****=====

1. 내 정보 보기
2. 내 정보 수정
3. 탈퇴하기
4. 농산물 메뉴로 가기
5. 로그아웃

메뉴 번호 입력 : 2

ID : rose
PWD : ro*****
NAME : 고객임

무엇을 변경하시겠습니까?

1. 비밀번호
2. 이름

메뉴 번호 입력 : 1

변경 내용 : rr

더 바꾸시겠습니까?(Y/N) : y

ID : rose
PWD : rr*****
NAME : 고객임

무엇을 변경하시겠습니까?

1. 비밀번호
2. 이름

메뉴 번호 입력 : 2

변경 내용 : 소비중

더 바꾸시겠습니까?(Y/N) : n

수정에 성공했습니다.

===== 일반 회원 메뉴 =====

1. 내 정보 보기
2. 내 정보 수정
3. 탈퇴하기
4. 농산물 메뉴로 가기
5. 로그아웃

메뉴 번호 입력 : 1

ID : rose
PWD : rr*****
NAME : 소비중

===== 일반 회원 메뉴 =====

1. 내 정보 보기
2. 내 정보 수정
3. 탈퇴하기
4. 농산물 메뉴로 가기
5. 로그아웃

메뉴 번호 입력 : 4

현재 KH마트 농산물 수량

복숭아 (과일) 2000원 : 150개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 100개

메뉴 번호 입력 : 3

구매 이력이 없습니다.

현재 KH마트 농산물 수량

복숭아 (과일) 2000원 : 150개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 100개

=====***** 고객 메뉴 *****=====

1. 농산물 구입하기
2. 농산물 환불하기
3. 구입한 농산물 보기
9. 메인으로 돌아가기

메뉴 번호 입력 : 1

복숭아 (과일) 2000원 : 150개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 100개

구매할 농산물 명을 입력하세요 : 복숭아

구매할 수량을 입력하세요 : 200

준비된 수량이 부족합니다.

더 진행하시겠습니까?(Y/N) : y

복숭아 (과일) 2000원 : 150개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 100개

구매할 농산물 명을 입력하세요 : 복숭아

구매할 수량을 입력하세요 : 10

복숭아 10개, 총 20000원입니다.

구매하시겠습니까?(Y/N) : y

구매 완료하였습니다.

더 진행하시겠습니까?(Y/N) : y

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 100개

구매할 농산물 명을 입력하세요 : 호두

구매할 수량을 입력하세요 : 100

호두 100개, 총 500원입니다.

구매하시겠습니까?(Y/N) : y

구매 완료하였습니다.

더 진행하시겠습니까?(Y/N) : y

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 2900개

오이 (채소) 1000원 : 100개

구매할 농산물 명을 입력하세요 : 오이

구매할 수량을 입력하세요 : 1

오이 1개, 총 1000원입니다.

구매하시겠습니까?(Y/N) : n

더 진행하시겠습니까?(Y/N) : n

현재 KH마트 농산물 수량

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 2900개

오이 (채소) 1000원 : 100개

===== 고객 메뉴 =====

1. 농산물 구입하기
2. 농산물 환불하기
3. 구입한 농산물 보기
9. 메인으로 돌아가기

메뉴 번호 입력 : 3

1. rose/2022-08-19/복숭아 2000원(개당) 10개, 총 20000원

2. rose/2022-08-19/호두 5원(개당) 100개, 총 500원

현재 KH마트 농산물 수량

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 2900개

오이 (채소) 1000원 : 100개

===== 고객 메뉴 =====

1. 농산물 구입하기
2. 농산물 환불하기
3. 구입한 농산물 보기
9. 메인으로 돌아가기

메뉴 번호 입력 : 2

1. rose/2022-08-19/복숭아 2000원(개당) 10개, 총 20000원

2. rose/2022-08-19/호두 5원(개당) 100개, 총 500원

환불할 번호를 입력하세요 : 0

잘못된 번호입니다.

현재 KH마트 농산물 수량

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 2900개

오이 (채소) 1000원 : 100개

===== 고객 메뉴 =====

1. 농산물 구입하기

2. 농산물 환불하기

3. 구입한 농산물 보기

9. 메인으로 돌아가기

메뉴 번호 입력 : 2

1. rose/2022-08-19/복숭아 2000원(개당) 10개, 총 20000원

2. rose/2022-08-19/호두 5원(개당) 100개, 총 500원

환불할 번호를 입력하세요 : 2

성공적으로 환불되었습니다.

현재 KH마트 농산물 수량

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 100개

===== 고객 메뉴 =====

1. 농산물 구입하기

2. 농산물 환불하기

3. 구입한 농산물 보기

9. 메인으로 돌아가기

메뉴 번호 입력 : 3

1. rose/2022-08-19/복숭아 2000원(개당) 10개, 총 20000원

현재 KH마트 농산물 수량

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 100개

===== 고객 메뉴 =====

1. 농산물 구입하기
2. 농산물 환불하기
3. 구입한 농산물 보기
9. 메인으로 돌아가기

메뉴 번호 입력 : 9

===== 일반 회원 메뉴 =====

1. 내 정보 보기
2. 내 정보 수정
3. 탈퇴하기
4. 농산물 메뉴로 가기
5. 로그아웃

메뉴 번호 입력 : 5

로그아웃합니다.

===== 메인 메뉴 =====

1. 로그인
2. 회원가입
9. 종료

메뉴 번호 입력 : 2

아이디 : mary

사용가능한 아이디입니다.

비밀번호 : mm

이름 : 고객임

당신은 관리자입니까?(Y/N) : n

일반 회원으로 회원가입이 진행됩니다.

회원가입이 성공하였습니다.

===== 메인 메뉴 =====

1. 로그인
2. 회원가입
9. 종료

메뉴 번호 입력 : 1

아이디 : mary

비밀번호 : mm

고객임님, 환영합니다.

===== 일반 회원 메뉴 =====

1. 내 정보 보기
2. 내 정보 수정
3. 탈퇴하기
4. 농산물 메뉴로 가기
5. 로그아웃

메뉴 번호 입력 : 4

현재 KH마트 농산물 수량

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 100개

===== 고객 메뉴 =====

1. 농산물 구입하기
2. 농산물 환불하기
3. 구입한 농산물 보기
9. 메인으로 돌아가기

메뉴 번호 입력 : 1

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 100개

구매할 농산물 명을 입력하세요 : 오이

구매할 수량을 입력하세요 : 99

오이 99개, 총 99000원입니다.

구매하시겠습니까?(Y/N) : y

구매 완료하였습니다.

더 진행하시겠습니까?(Y/N) : n

현재 KH마트 농산물 수량

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 1개

===== 고객 메뉴 =====

1. 농산물 구입하기
2. 농산물 환불하기
3. 구입한 농산물 보기
9. 메인으로 돌아가기

메뉴 번호 입력 : 3

1. mary/2022-08-19/오이 1000원(개당) 99개, 총 99000원

현재 KH마트 농산물 수량

복숭아 (과일) 2000원 : 140개

호두 (견과) 5원 : 3000개

오이 (채소) 1000원 : 1개

===== 고객 메뉴 =====

1. 농산물 구입하기
2. 농산물 환불하기
3. 구입한 농산물 보기
9. 메인으로 돌아가기

메뉴 번호 입력 : 9

===== 일반 회원 메뉴 =====

1. 내 정보 보기
2. 내 정보 수정
3. 탈퇴하기
4. 농산물 메뉴로 가기
5. 로그아웃

메뉴 번호 입력 : 5

로그아웃합니다.

=====***** 메인 메뉴 *****=====

1. 로그인
2. 회원가입
9. 종료

메뉴 번호 입력 : 1

아이디 : admin

비밀번호 : ad

나관리님, 환영합니다.

=====***** 직원 메뉴 *****=====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 5

1. rose/2022-08-19/복숭아 2000원(개당) 10개, 총 20000원
 2. mary/2022-08-19/오이 1000원(개당) 99개, 총 99000원
- 총 매출 119000원입니다.

=====***** 직원 메뉴 *****=====

1. 새 농산물 추가
2. 농산물 수정
3. 농산물 삭제
4. 농산물 목록
5. 매출 보기
6. 로그아웃

메뉴 번호 입력 : 6

로그아웃합니다.

=====***** 메인 메뉴 *****=====

1. 로그인
2. 회원가입
9. 종료

메뉴 번호 입력 : 1

아이디 : mary

비밀번호 : mm

고객임님, 환영합니다.

=====***** 일반 회원 메뉴 *****=====

1. 내 정보 보기
2. 내 정보 수정
3. 탈퇴하기
4. 농산물 메뉴로 가기
5. 로그아웃

메뉴 번호 입력 : 3

정말 탈퇴하시겠습니까?(Y/N) : k

잘못 입력하셨습니다.

정말 탈퇴하시겠습니까?(Y/N) : n

=====***** 일반 회원 메뉴 *****=====

1. 내 정보 보기
2. 내 정보 수정
3. 탈퇴하기
4. 농산물 메뉴로 가기
5. 로그아웃

메뉴 번호 입력 : 3

정말 탈퇴하시겠습니까?(Y/N) : y

탈퇴 성공했습니다.

로그아웃합니다.

=====***** 메인 메뉴 *****=====

1. 로그인
2. 회원가입
9. 종료

메뉴 번호 입력 : 1

아이디 : mary

비밀번호 : mm

존재하지 않는 회원입니다.

===== 메인 메뉴 =====

1. 로그인
2. 회원가입
9. 종료

메뉴 번호 입력 : 9

프로그램 종료