* class naming
* 공통적으로 사용하거나 크게 사용하는 모델의 클래스 이름을 정한다. 분석/설계 과정에서 class diagram을 대체한다

* Model(Vo,Dao,Service)
* Vo

1. 관리
   * MemberVo / MemberInfoVo (회원) -- has a 관계
   * CropsVo / CropsInfoVo (작물) -- has a 관계
   * AccusePostVo (신고게시판)

1. 구매
   * TradePostVo (거래 게시판)
   * TradeCommentVo (댓글) -> 할예정
   * RentVo (밭대여신청)
   * HarvestVo (수확상태)
   * BuyerScheduleVo (대여일정)
   * ApplySellerVo(판매자 신청)

1. 판매
   * FarmVo (밭)
   * RecruitVo (모집)
   * SellerScheduleVo (대여일정)

* Dao

1. 관리자
   * MemberDao
   * CropsDao
   * AccusePostDao
2. 구매자
   * TradePostDao
   * RentDao
   * BuyerFarmDao
   * BuyerScheduleDao
3. 판매자
   * SellerFarmDao
   * RecruitDao
   * SellerScheduleDao

* Service

1. 구매자
   * RentService -구매자가 농지를 대여하는 일련의 서비스를 정의
   * TradeService - 구매자가 농작물을 판매하거나 교환하는 서비스를 정의
   * BuyerFarmService - 구매자가 대여한 밭의 관리를 위한 서비스
2. 관리자, 공통
   * CommonService - 작물 사전 열람, 서비스의 기본 정보(기본정보를 추후에 DB에 입력하는 버전 시)가 변경될 시 처리하는 서비스
   * MemberService - 회원의 정보를 관리하기 위한 서비스. 회원가입,회원정보수정(판매자,구매자측 모두),회원탈퇴
   * AccuseService - 회원들의 신고 게시물들의 처리를 위한 서비스. 등록, 삭제, 관리자의 승인, 거절
   * AdminService - 관리자가 회원들의 정보를 관리자의 권한으로 수정하는 서비스 (판매자 승인, 회원정지, 강제탈퇴)
3. 판매자
   * SellerFarmService - 판매자의 농지를 등록, 수정하는 서비스를 처리, 농지를 대여중인 구매자들을 관리하는 서비스(대여자 관리는 추후에 나눠야 할 상황에 나누기)
   * RecruitService - 등록한 농지의 구매자를 모집하는 서비스

* Controller

1. 공통
   * HomeController(login, register, logout, home화면,dictionary)
2. 관리자
   * AdminController(관리자 뷰)
3. 구매자
   * BuyerController(구매자 뷰)
4. 판매자
   * SellerController(판매자 뷰)

* target url & Jsp & Folder Naming Rule
* Controller Target Url ( .do )
* Camel Case
  + - 수행되여야 할 기능 메소드명과 같거나 비슷하게 짓는다.

* Jsp File
* Snake Case
  + /seller/farm/(밭을 대여중인 멤버들. 이름 뽑아와서 출력하는 View

* Folder
  + 예) /home/memberManagement/update\_member.jsp