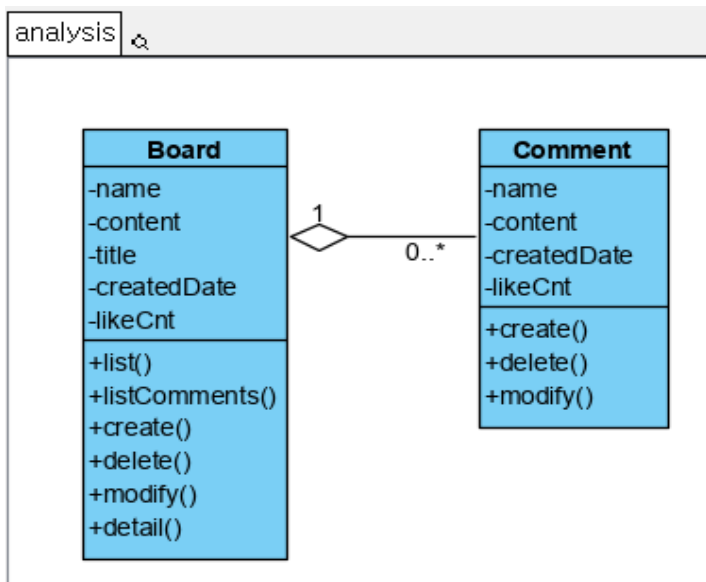
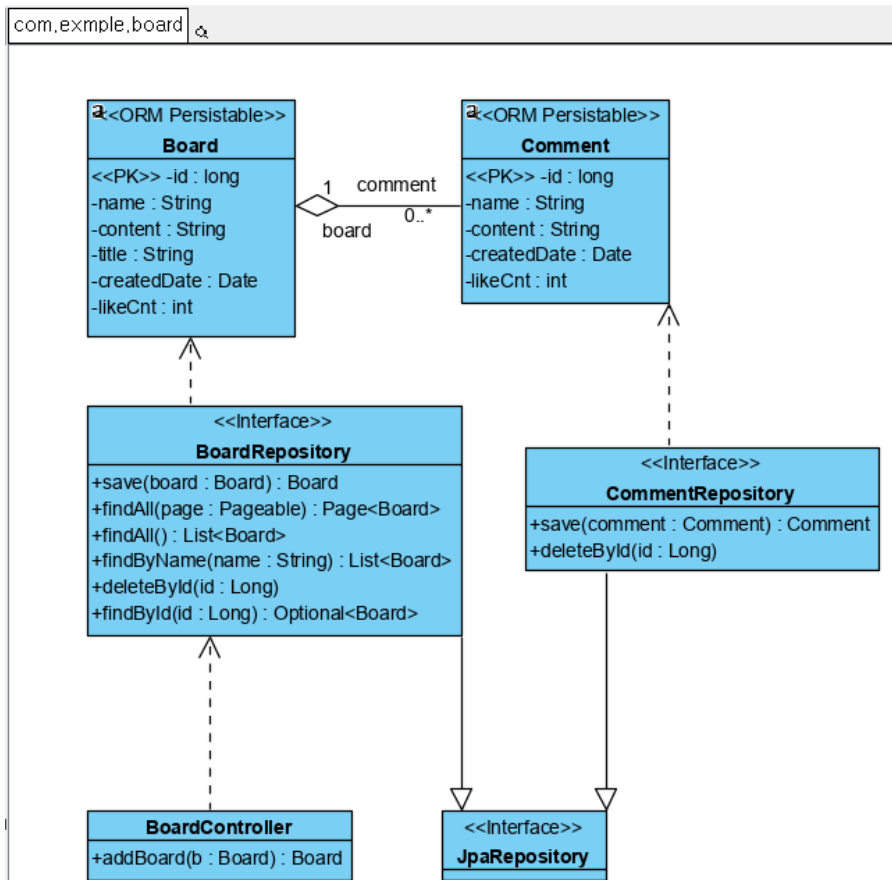


분석 - Analysis Object Model



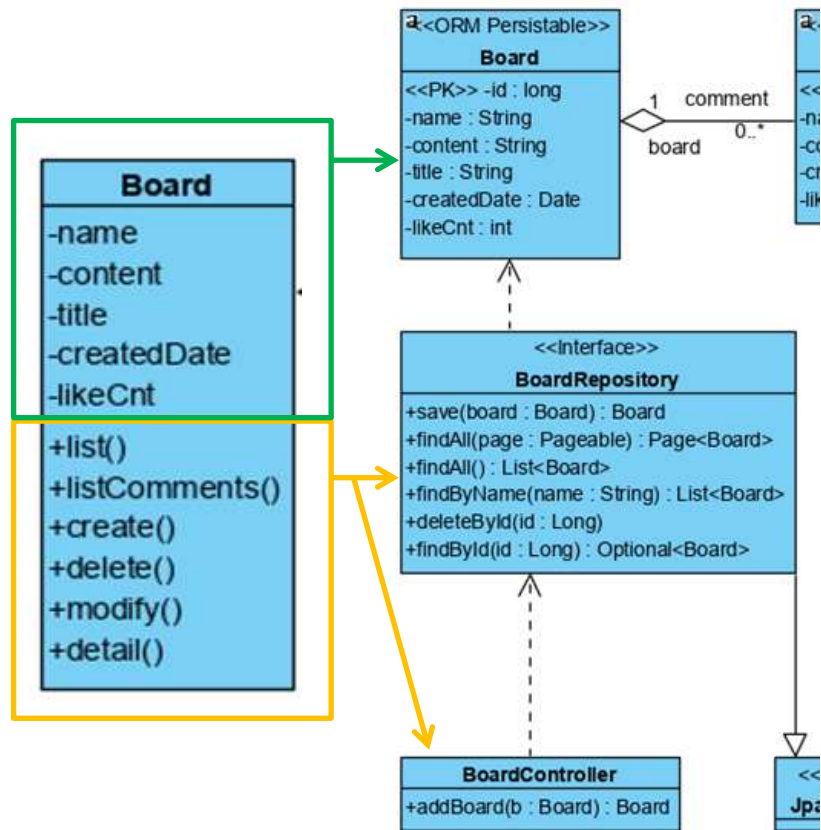
- ✓ Analysis Object Model은 특정 개발 플랫폼에 종속 받지 않는다.
- ✓ UseCase 개념모델로부터 객체 모델을 생성
 1. UseCase에서 도출 가능한 도메인 객체(Object), 표현
 2. UseCase에서 도출 가능한 속성과 메소드 작성
 - ❖ 속성과 메소드에는 타입을 작성하지 않음
 3. 객체간 관계 표현

설계 – Class Diagram



- ✓ 개발 플랫폼의 설계 패턴에 맞도록 표현 한다
- ✓ 타입에 대한 명시
- ✓ 명명 규칙은 플랫폼의 규칙을 따름
- ✓ 필요한 경우 dependency 관계에 대해서 표현

분석 모델 -> 설계 모델



- Spring Framework의 Entity, Repository, Controller 패턴에 맞게 클래스 분해
- 객체 속성은 Entity 클래스에 표현
- 객체 오퍼레이션은 Repository 클래스에 표현
 - 추가적인 비즈니스 로직(검증 및 객체관계 처리) 이 있는 경우 API영역(외부접근)과 Data Access 영역을 분리하여 Controller와 Repository로 표현
 - 실제 개발환경에서의 비즈니스 로직 중 트랜잭션 처리는 Service 클래스에 작성한다.

<https://www.logicbig.com/tutorials/spring-framework/spring-data/transactions.html>