

# 유스케이스 다이어그램 (Usecase-Diagram)

윤종두

# UML

분류	다이아그램 유형		목적
구조 다이어그램 (structure diagram)	클래스 다이어그램 (class diagram)		시스템을 구성하는 클래스들 사이의 관계를 표현한다.
	객체 다이어그램 (object diagram)		객체 정보를 보여준다.
	복합체 구조 다이어그램 (composite structure diagram)		복합 구조의 클래스와 컴포넌트 내부 구조를 표현한다.
	배치 다이어그램 (deployment diagram)		소프트웨어, 하드웨어, 네트워크를 포함한 실행 시스템의 물리 구조를 표현한다.
	컴포넌트 다이어그램 (component diagram)		컴포넌트 구조 사이의 관계를 표현한다.
	패키지 다이어그램 (package diagram)		클래스나 유즈 케이스 등을 포함한 여러 모델 요소들을 그룹화해 패키지를 구성하고 패키지들 사이의 관계를 표현한다.
행위 다이어그램 (behavior diagram)	활동 다이어그램 (activity diagram)		업무 처리 과정이나 연산이 수행되는 과정을 표현한다.
	상태 머신 다이어그램 (state machine diagram)		객체의 생명주기를 표현한다.
	유즈 케이스 다이어그램 (use case diagram)		사용자 관점에서 시스템 행위를 표현한다.
	상호작용 다이어그램 (interaction diagram)	순차 다이어그램 (sequence diagram)	시간 흐름에 따른 객체 사이의 상호작용을 표현한다.
		상호작용 개요 다이어그램 (interaction overview diagram)	여러 상호작용 다이어그램 사이의 제어 흐름을 표현한다.
		통신 다이어그램 (communication diagram)	객체 사이의 관계를 중심으로 상호작용을 표현한다.
		타이밍 다이어그램 (timing diagram)	객체 상태 변화와 시간 제약을 명시적으로 표현한다.

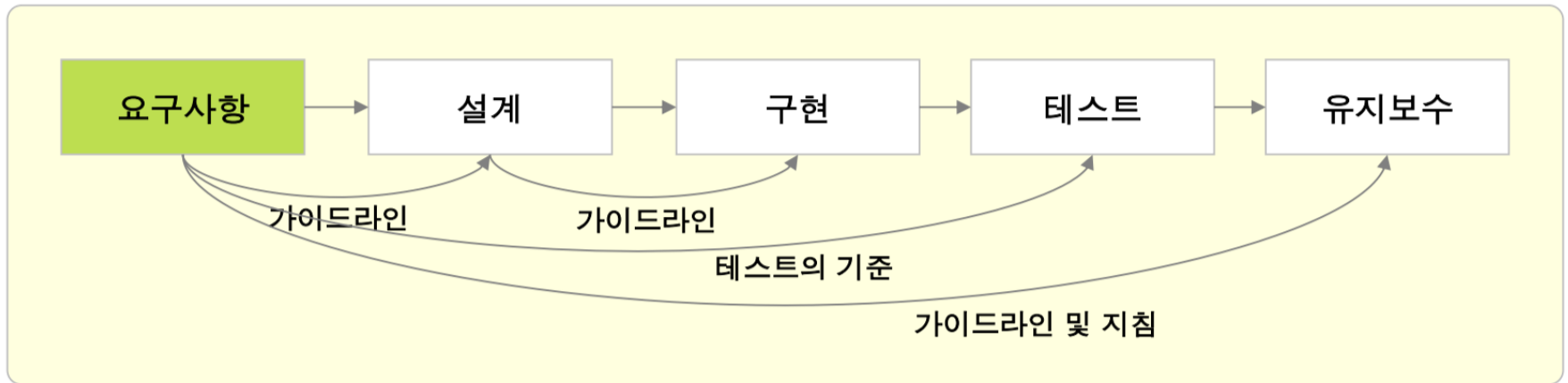
# 유스케이스 다이어그램 이란?

- 사용자(Actor) 중심의 시스템 구상
- SW의 기능적 요구사항 분석
- 사용자의 시각에서 시스템의 범위와 기능을 정의한 모델
- 사용자가 어떤 기능을 사용할 수 있는지 알 수 있음

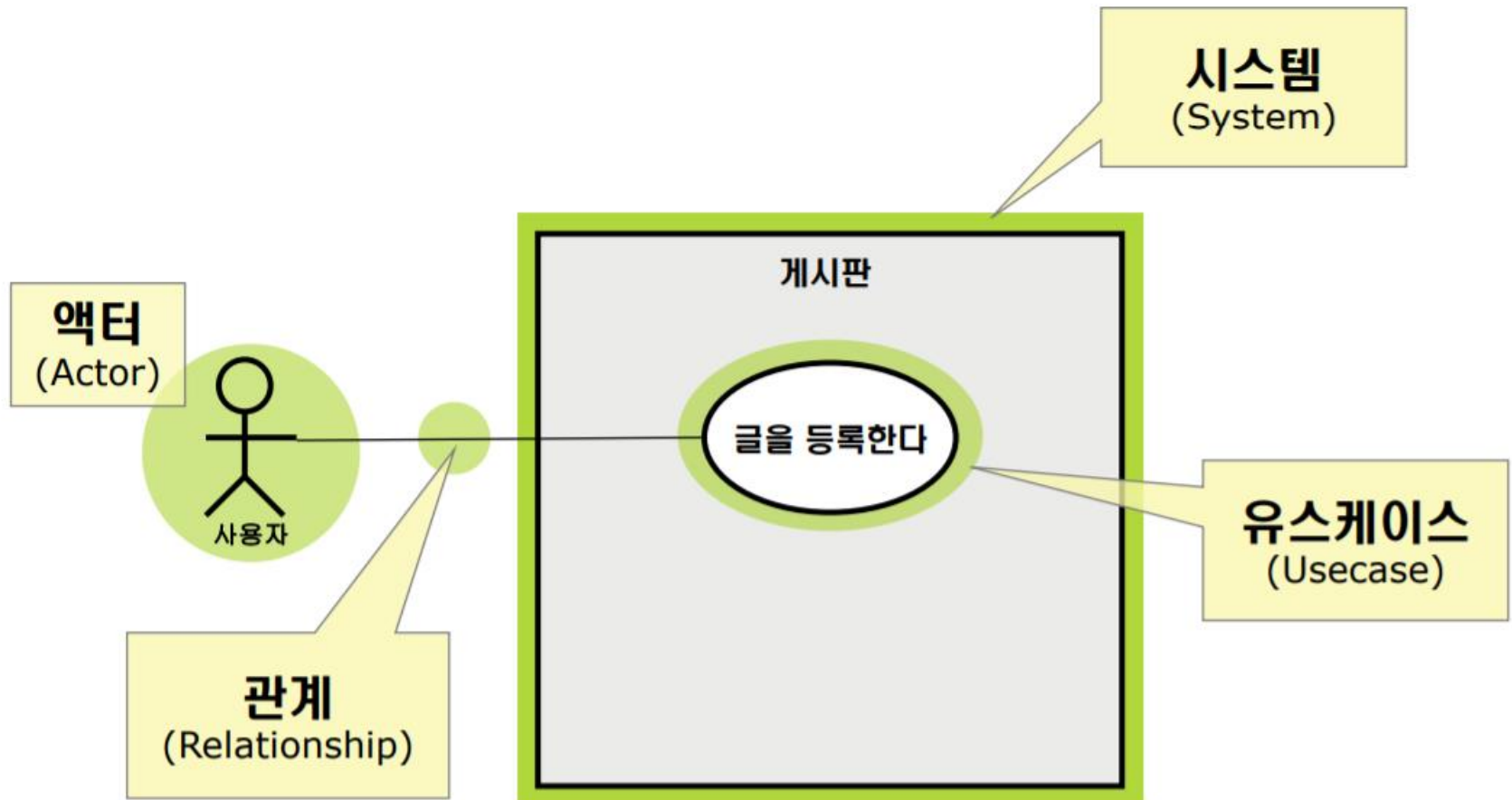
# 작성시기

- 소프트웨어의 요구사항을 분석/정의할 때
- 프로젝트 개발범위를 정의할 때
- 소프트웨어의 세부기능을 분석할 때

# 요구사항 분석의 중요성



# 예시

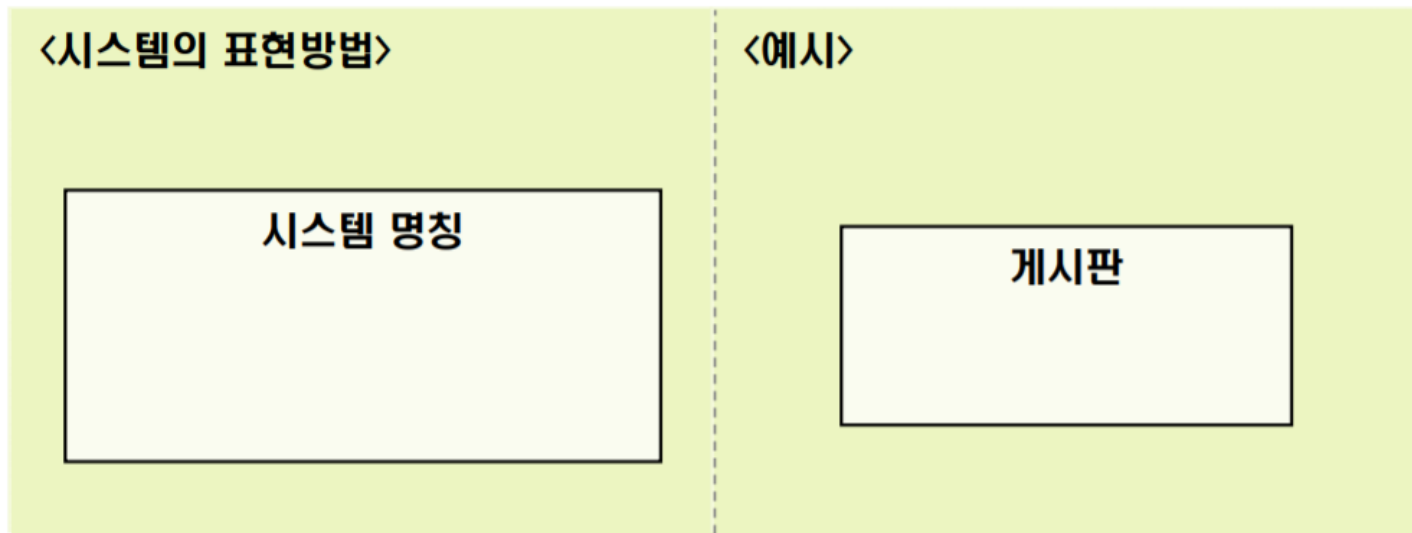


# 구성요소

- 시스템(System, Scope)
- 유스케이스(Usecase)
- 액터(Actor)
- 관계(Relation)

# 구성요소

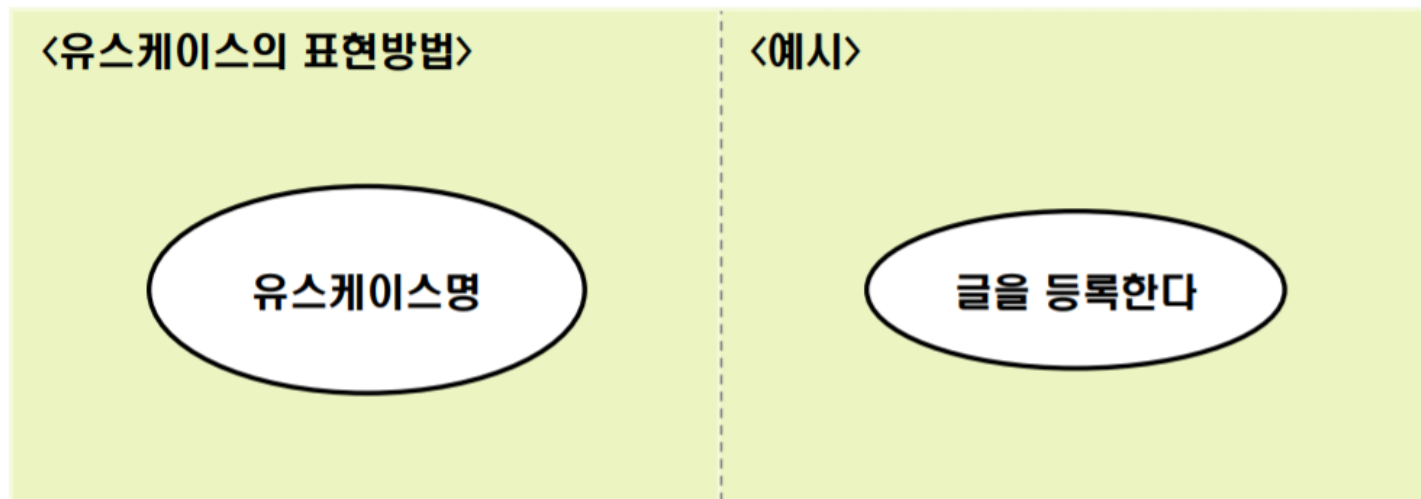
- 시스템(system, scope)
  - 개발하고자 하는 시스템
  - 사각형으로 표현
  - 시스템 명칭을 사각형 안쪽 상단에 기술





# 구성요소

- 유스케이스(usecase)
  - 시스템이 제공하는 기능
  - 동그라미로 표현
  - 이름은 "~한다"와 같이 동사로 표현



# 구성요소

- 액터(Actor)
  - 시스템 외부에 있으면서 상호 작용을 하는 사람(Primary) 또는 다른 시스템(Secondary)
  - 액터의 역할로 이름을 정함

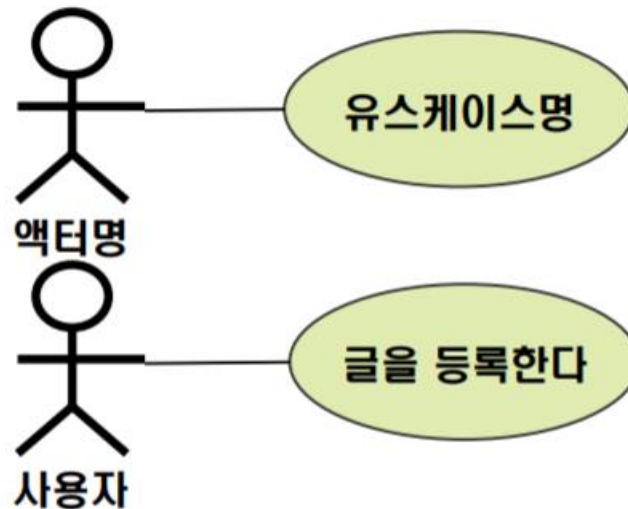


# 구성요소

- 관계(Relation)
  - 액터와 유스케이스 사이의 의미있는 관계
  - 종류
    - 연관 관계(association)
    - 의존 관계(dependency)
      - 포함 관계(include)
      - 확장 관계(extends)
    - 일반화 관계(generalization)

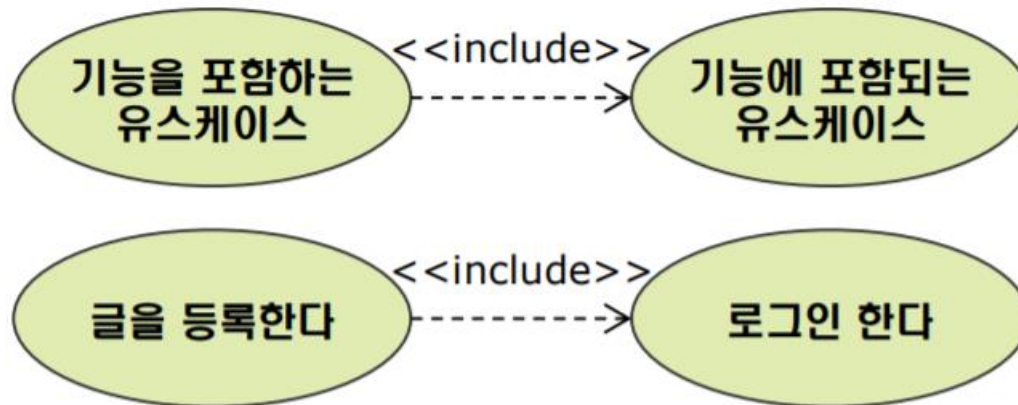
# 구성요소

- 연관관계(association)
  - 유스케이스와 액터간의 상호작용 표현
  - 유스케이스와 액터를 실선으로 연결



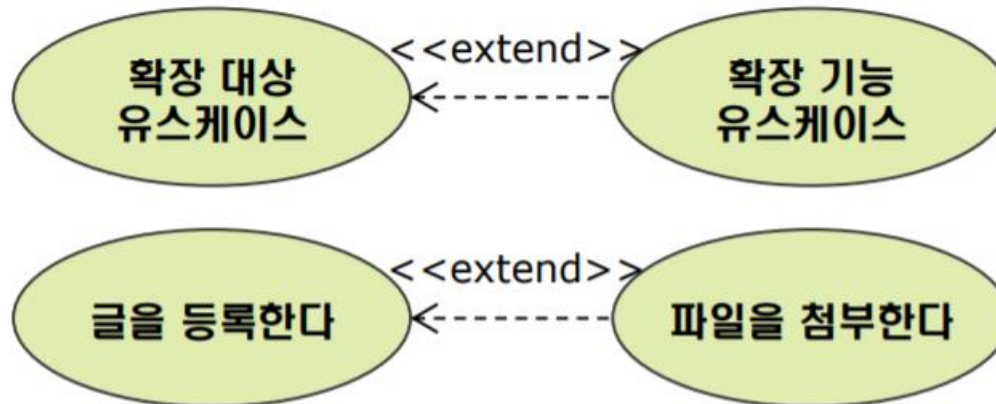
# 구성요소

- 포함관계(include)
  - 포함되는 유스케이스는 포함하는 유스케이스를 실행하기 위해 반드시 실행되어야 하는 경우 적용



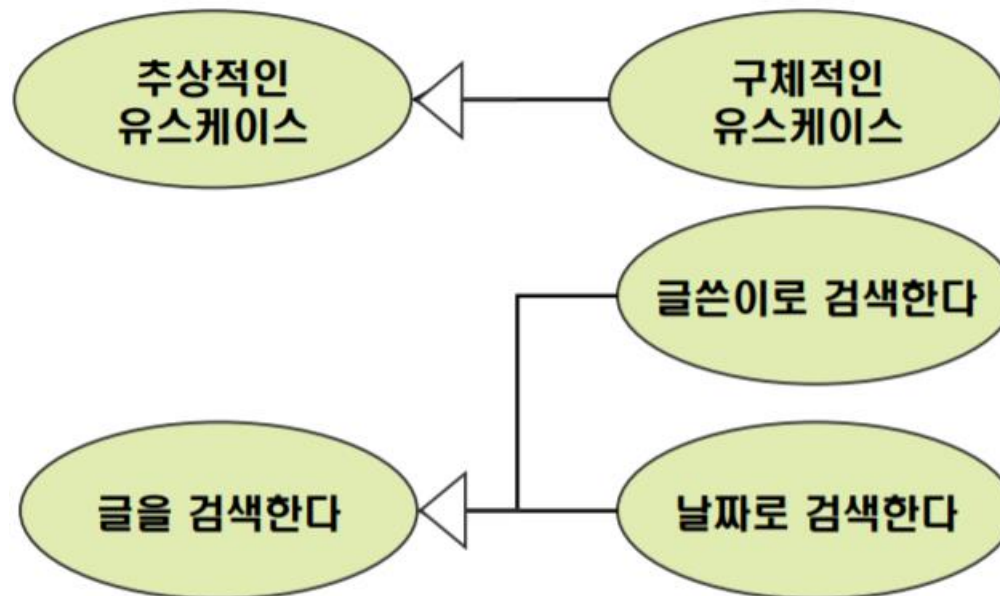
# 구성요소

- 확장관계(extends)
  - 확장 대상 유스케이스를 수행 할 때에 특정 조건에 따라 확장 가능 유스케이스를 수행하기도 하는 경우 적용



# 구성요소

- 일반화 관계(generalization)
  - 유사한 유스케이스(액터)들 을 모아 그들을 추상화한 유스케이스(액터)와 연결 (is-a)
  - 그룹핑을 통해 이해도를 높이기 위한 관계



# 관계 식별

- 연관 관계
  - 액터와 유스케이스 간에 상호작용이 존재하는가?
- 포함 관계
  - 이 유스케이스를 실행하기 위하여 반드시 실행되어야 하는 유스케이스가 존재하는가?
- 확장관계
  - 이 유스케이스를 실행 함으로써 선택적으로 실행되는 유스케이스가 있는가?
- 일반화 관계
  - 액터 또는 유스케이스가 구체화된 다른 여러 액터나 유스케이스를 가지고 있는거?



# 작성 순서

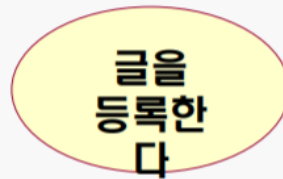
1

액터 식별



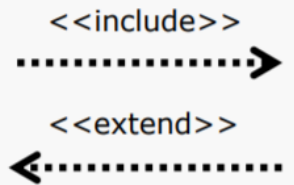
2

Use Case 식별



3

관계 정의



# 예시

- 글을 등록, 수정, 삭제 할 수 있는 게시판을 개발한다.
  - 글을 등록할 때에는 파일을 첨부할 수 있다.
  - 글을 조회하여 읽을 수 있다.
  - 등록된 글은 글쓴이 혹은 날짜 별로 검색할 수 있다.
  - 게시판의 모든 기능은 사용자 로그인 후에 사용할 수 있다.

# 예시

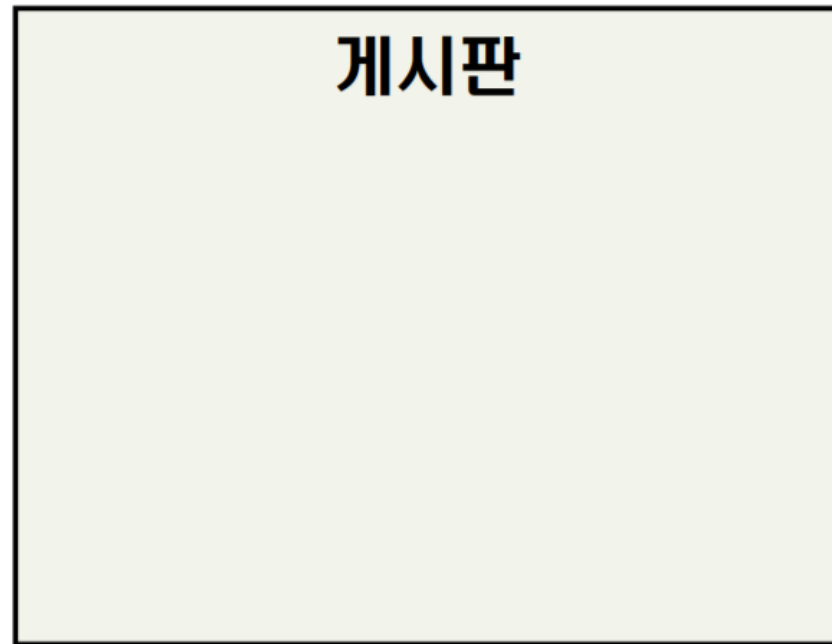
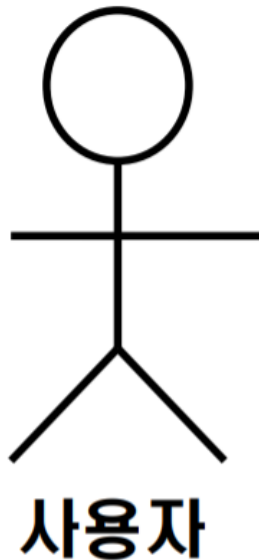
- 시스템 식별
  - 요구사항을 통해 만들고자 하는 시스템은 “게시판” 이다.



게시판

# 예시

- 액터 식별
  - 외부에서 상호작용하는 액터로 글을 등록하고 삭제하는 등의 역할을 하는 "사용자"가 식별됨



# 예시

- 유스케이스 식별
  - 게시판을 통해 글을 등록, 수정, 조회하는 등의 작업을 할 수 있음

**글을 등록한다**

**글을 수정한다**

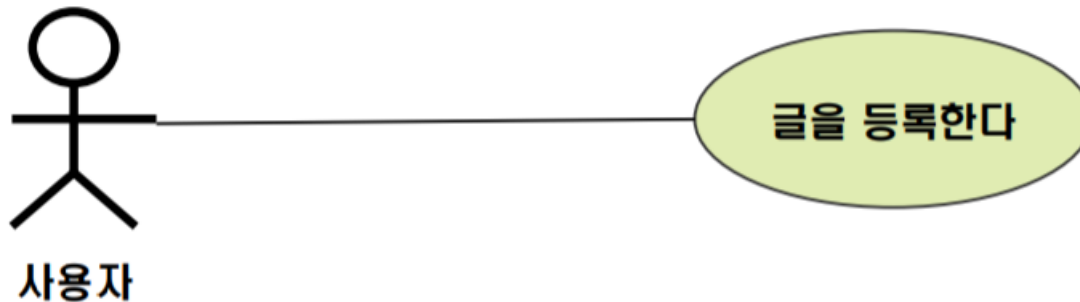
**글을 조회한다**

**글을 삭제한다**

**글을 검색한다**

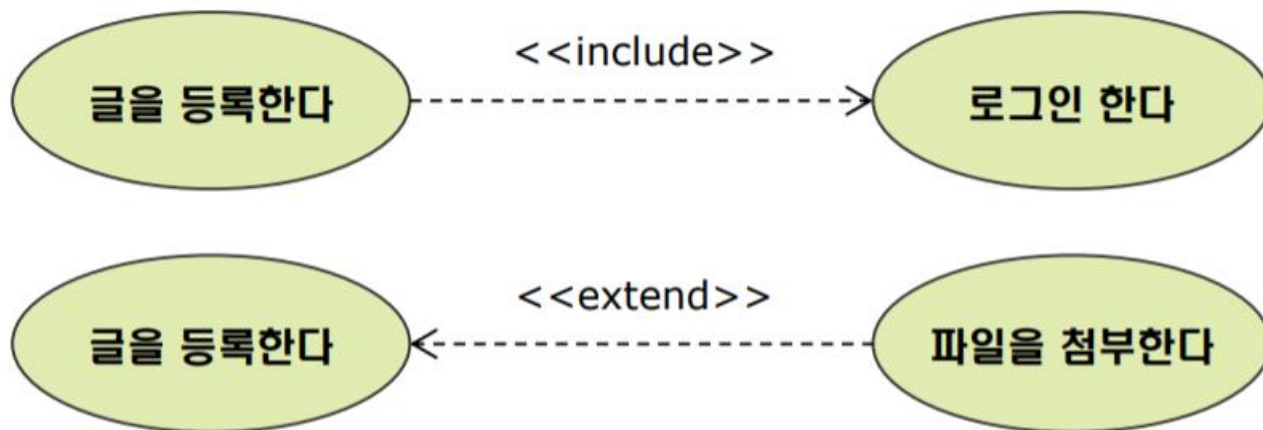
# 예시

- 관계 정의
  - 연관관계
    - 액터 – 유스케이스
    - 예 : 사용자가 글을 등록한다.



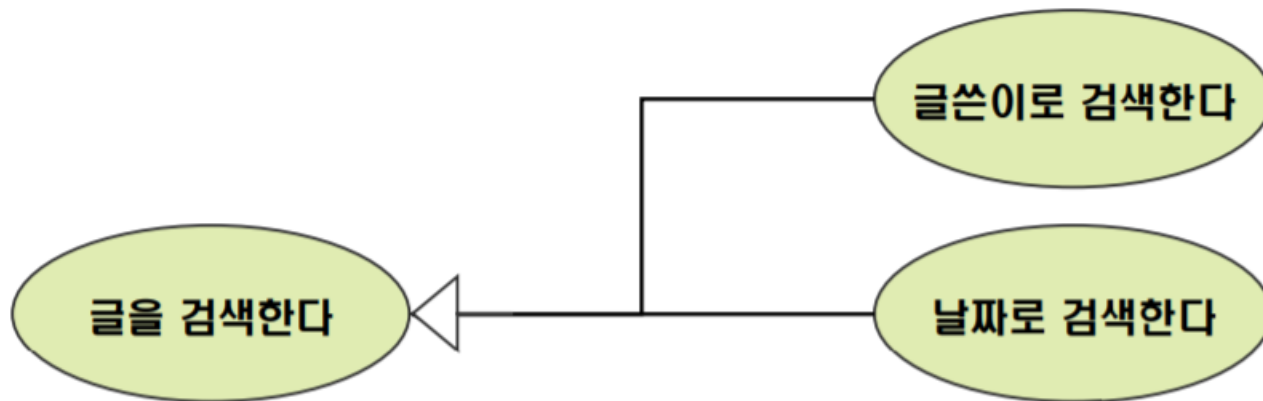
# 예시

- 관계 정의
  - 포함관계, 확장관계
    - 유스케이스 - 유스케이스
    - 예 : 글을 등록할때는 로그인 이 필요하다.
    - 예 : 글을 등록할때는 파일을 등록할 수 있다.



# 예시

- 관계 정의
  - 일반화 관계
    - 유스케이스-유스케이스, 액터-액터
    - 글 검색 : 글쓴이로 검색, 날짜로 검색





# 예시

