

I. UML 의 정의

1. UML(United Modeling Language)

- 모델을 만드는 통합 언어

2. Model

- 실제 무언가를 만들기 전에 검증해 보는 것
- "이렇게 만들면 잘 동작하고 비용대비 효율성이 있을까?" 하고 따져보는 것
- 모델이 입은 옷을 보며 내가 있었을 경우를 상상해 보고 구매를 결정하는 것과 같은 이치



I. UML 의 정의

3. 주의사항

- 개 집을 짓는데 설계도를 작성하지는 않는다.
- 설계를 위한 설계는 안 하느니만 못하다.

4. UML 을 사용하는 목적

- 다른 사람들과의 의사소통 또는 설계 논의
- 전체 시스템의 구조 및 클래스의 의존성 파악
- 유지보수를 위한 설계의 back-end 문서

=> 설계 내용을 공유하고 유지보수를 위해 필요하다.

I. UML 의 정의

5. UML 의 종류

- 구조 다이어그램 7개, 행위 다이어그램 7개로 총 14종류의 다이어그램이 있다.
 - 구조 다이어그램 : 시스템의 개념, 관계 등의 측면에서 **요소**들을 나타내냄(정적)
 - 행위 다이어그램 : 요소들간의 변화나 흐름, 주고받는 데이터 등의 **동작**을 나타냄(동적)

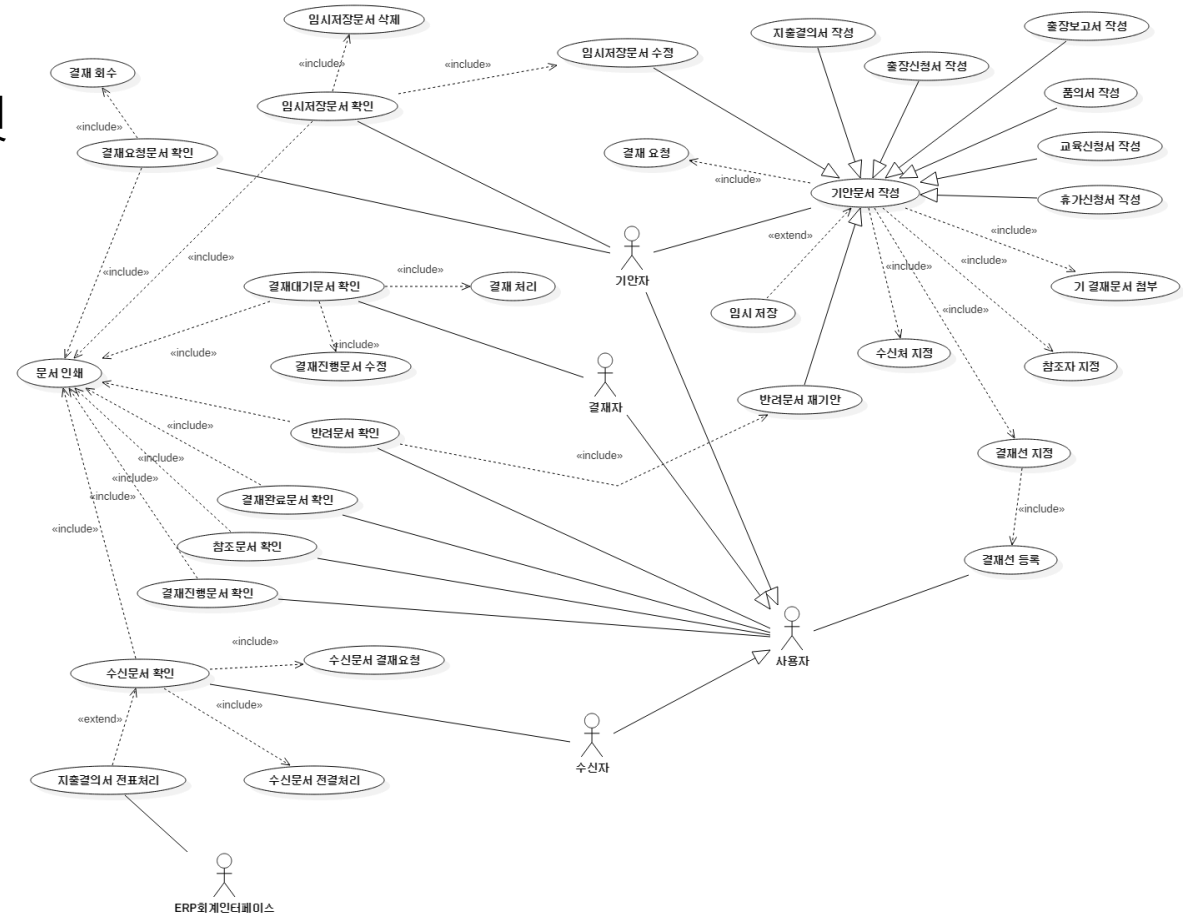
방식	특징
Class Diagram	클래스와 그들간의 관계를 정의한 Diagram
Use Case Diagram	사용자 관점에서 논리적인 시스템의 서비스 혹은 Use Case 를 정의
Sequence Diagram	특정 프로세스의 진행의 흐름을 정의

II. Use Case Diagram

1. Use Case Diagram

- 사용자 입장에서 본 시스템 동작에 대한 시나리오
- 여기서 중요한 점은 사용자의 관점에서 작성 된다는 것

명칭	설명
Actor	시스템의 사용자
Use case	액터가 사용 하는 기능
Generalize	상속 관계
Include	호출, 특정 Use case 의 하위작업의 관계를 나타냄
Extend	include 와 비슷함, 선택적인 하위 작업 관계
경계	하나의 시스템을 구분하기 위한 영역



II. Use Case Diagram

2. Actor

- 시스템의 사용자, 운영자 또는 해당 시스템을 사용하는 외부 시스템
- 사람 모양으로 표시 된다.

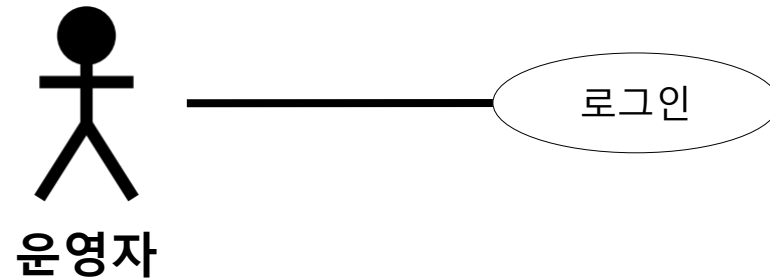


사용자, 운영자, 외부시스템

II. Use Case Diagram

3. Use Case

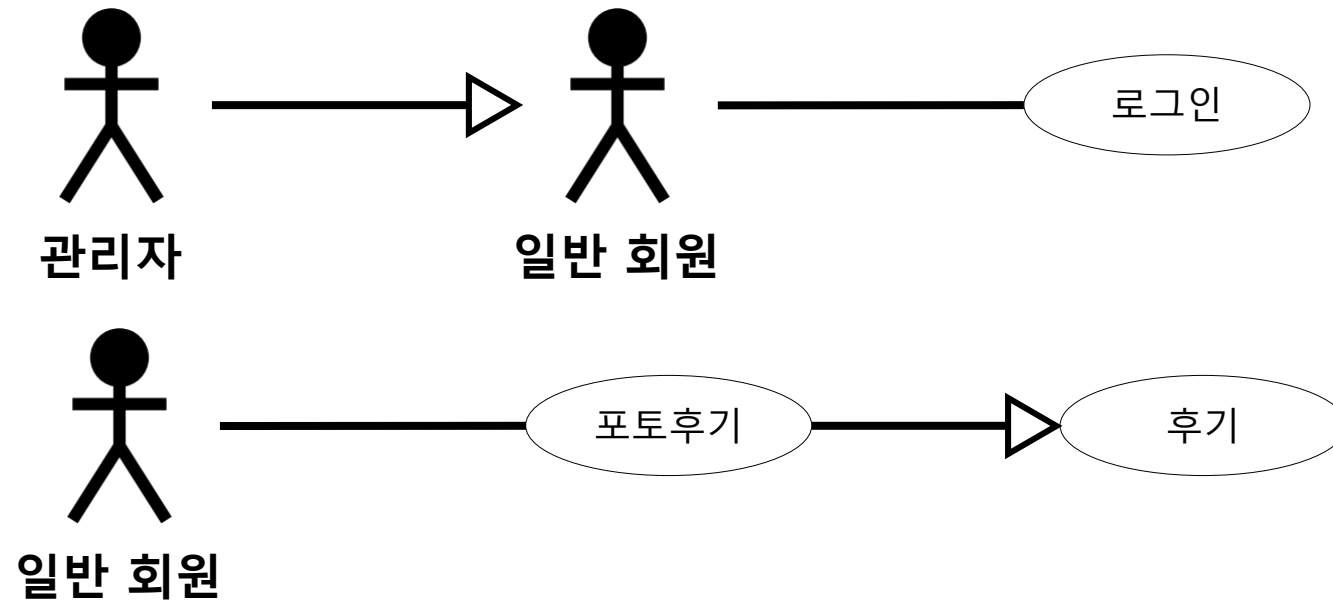
- Actor 가 사용하는 기능
- 타원으로 표시 된다.
- 아래는 사용자가 로그인 기능을 이용하는 것을 표현한 다이어그램



II. Use Case Diagram

4. Generalize

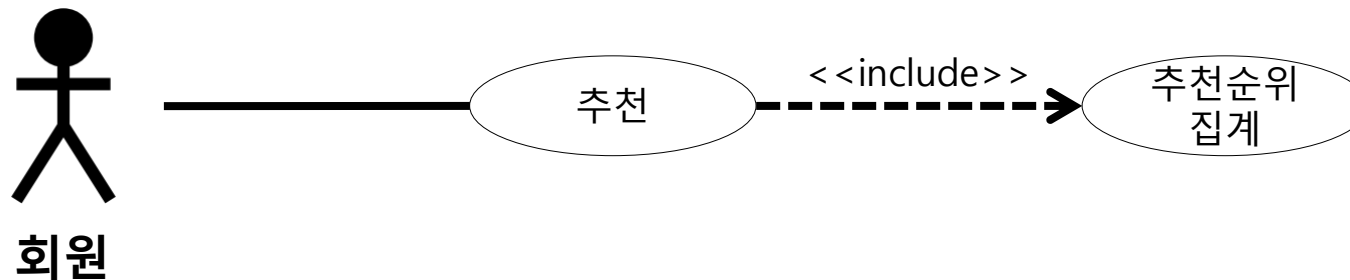
- 일반화, 일반화의 오류?
- JAVA 를 배운 사람들은 딱딱한 사람들이 많으니 너도 딱딱한 성격 이겠네?
- **재가 하니깐 너도 할 수 있어!**
- 아래는 관리자가 일반회원으로 일반화 하여 로그인 기능을 이용 할 수 있다.(상속 개념)



II. Use Case Diagram

5. Include

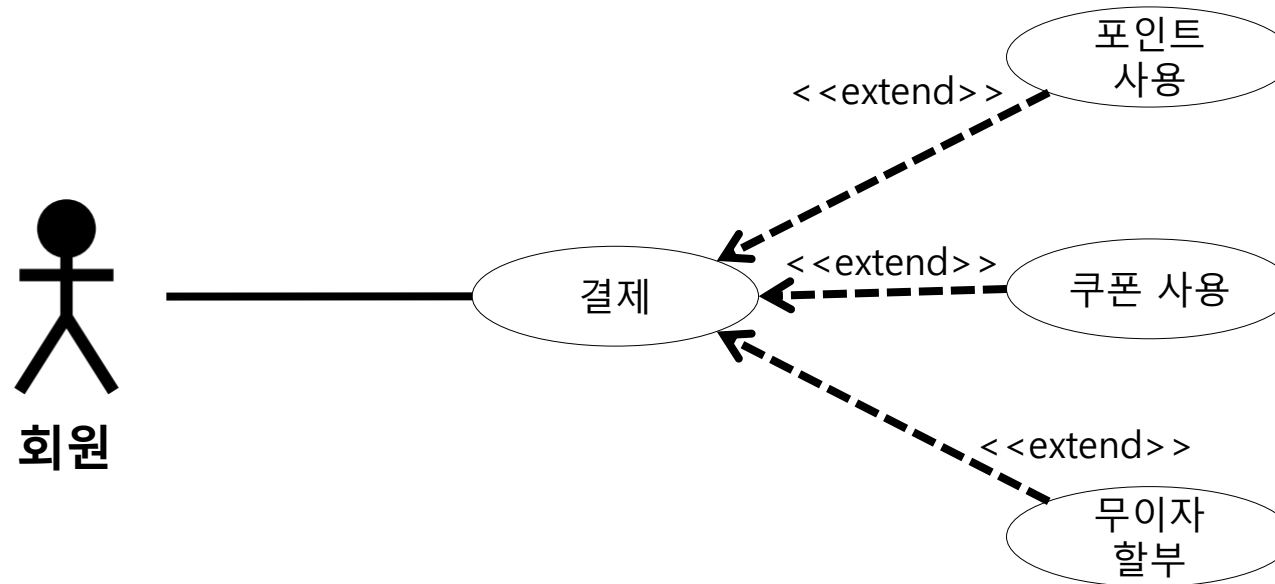
- 호출,포함의 의미
- 어떤 기능을 실행 할 때 호출(포함)되는 다른 기능들을 의미 한다.
- 아래는 추천 기능을 실행 했을 경우 추천수를 통한 추천 순위를 집계하는 기능이 실행 된다.



II. Use Case Diagram

6. Extend

- 확장, 어떤 기능을 이용 할 때 사용 기능을 확장 하는 것이다.
- 확장된 기능들은 필수가 아니라 선택이다.



II. Use Case Diagram

7. 경계

- 경계는 시스템 내부와 외부로 구분하는 사각형 이다.
- 주로 복수의 시스템이 연동되는 경우 사용 한다.

