

# UML

---

- 소프트웨어 개발 주기 전체 지원
- 다양한 diagram 지원
- 요구사항 분석 단계에서 use case diagram을 많이 사용
- domain concept model/software design을 기술할 때 Class diagram을 많이 사용

## Use Case

---

- 시스템을 사용하는 사용자의 입장이 핵심
- 공통적인 사용자 목표에 의해 묶인 시나리오들의 집합
- 내용 전달이 잘 되어야 함
- 시스템 Use case : 소프트웨어 시스템과의 상호작용 표현
- 비즈니스 Use case : 비즈니스와 고객 또는 이벤트와의 상호작용 표현
- Flow of Events : 사용자와 시스템 간의 상호작용을 서술하는 일련의 스텝들 -> 발생가능한 Sequence에 대해, Use Case를 시작시킨 행위자는 왼쪽, 결과를 받는 행위자는 오른쪽에 표현

## Actor

---

- 여러 사람 -> 하나의 액터, 하나의 사람 -> 여러 액터로 매핑 가능
- Use Case를 시작시키고, 끝나면 결과를 받음
- Primary Actor : 주 목적을 가지고 주도적으로 접근
- Secondary Actor : Use Case를 실행하기 위해 시스템이 필요로 하는 지원 액터

## Use Case Diagram

---

- 시스템의 설계도
- Use Case들의 목차의 그래픽적인 테이블 표현
- Actor / Use Case / 이들의 관계로 구성
- 시스템의 사용에 대한 시나리오의 집합
- 사용자의 관점에서 시스템을 모델링(사용자가 시스템에 바라는 바)

## System

- Use Case를 둘러싸는 사각형으로 표현(행위자는 시스템 외부에, Use Case는 내부에)

## Extend

- 하나의 Use Case가 다른 Use Case를 Extend하는 관계 -> Extension Point : Extend할 Use Case로 가는 행동타입, Condition : Extend할 Use Case로 가는 트리거
- 기존의 유즈케이스에 추가적으로 하는 행동이 있는 경우, 독립적으로 정리됨

## Include

- Use Case가 다른 Use Case에 include됨
  - 포함하는 관계 : include되는 Use Case가 include 하는 Use Case에 속함

## Use Case Description

---

- 이름 (Name)
- Actor 기술 (Description of Actor)
  - 요약 (Summary, Brief Description)
  - 역할 (Role)
- Usecase 기술 (Description of Usecases)
  - 요약 (Summary, Brief Description)
  - 사전조건 (Pre-condition)
  - 사후조건 (Post-condition)
  - 주 사건 흐름 (Main flow of event)
  - 대안 흐름 (Alternative flow)
  - 예외 흐름 (Exception flow)
  - 시나리오 (Scenario)
  - 기타
    - 비기능적 요구사항(Non-Functional Requirements)
    - 비즈니스 룰 (Business Rule)
    - 사용자 화면 (User Interface)