

# 소프트웨어공학 활용

## UML



한국기술교육대학교  
온라인평생교육원

## 학습내용

- UML 개요
- UML 실습

## 학습목표

- UML의 기본 개념과 종류를 이해하고 설명할 수 있다.
- UML 도구를 활용하여 클래스를 설계하고 설명할 수 있다.

# UML 개요

## 1 UML이란?

### 1 UML 정의



UML

Unified Modeling Language의 약자로  
통합 모델링 언어



- 업무, 소프트웨어 애플리케이션, 시스템 아키텍처를 모델링 하는데 사용되는 **표준 비주얼 모델링 언어**



UML은 프로그래밍 언어가 아니라 도구 UML 다이어그램을 사용하여 **다양한 언어코드를 생성**하는데 사용됨

## 2 UML 목표

01

사용자들에게  
쉽고 다양하게  
표현할 수 있는  
**모델링 언어를 제공**

02

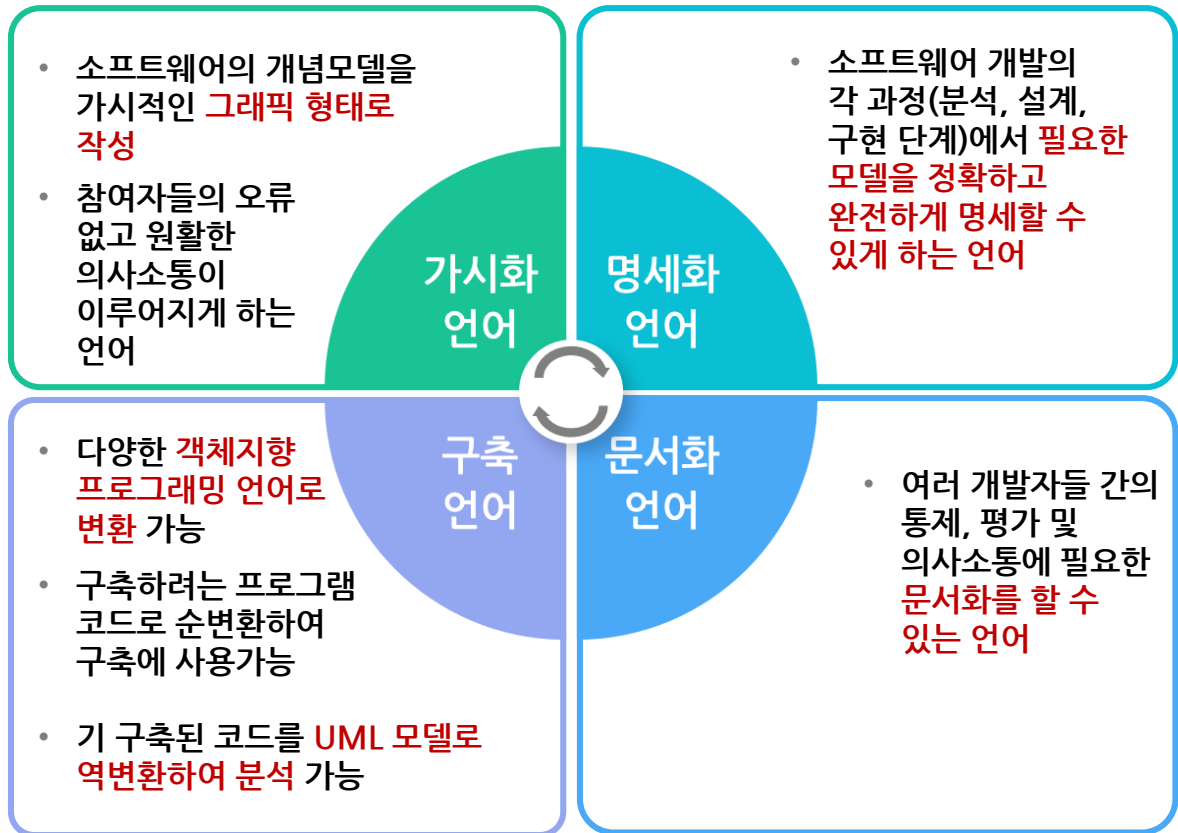
개발언어 및 개발  
프로세스와 독립적인  
**Specification을 제공**

# UML 개요



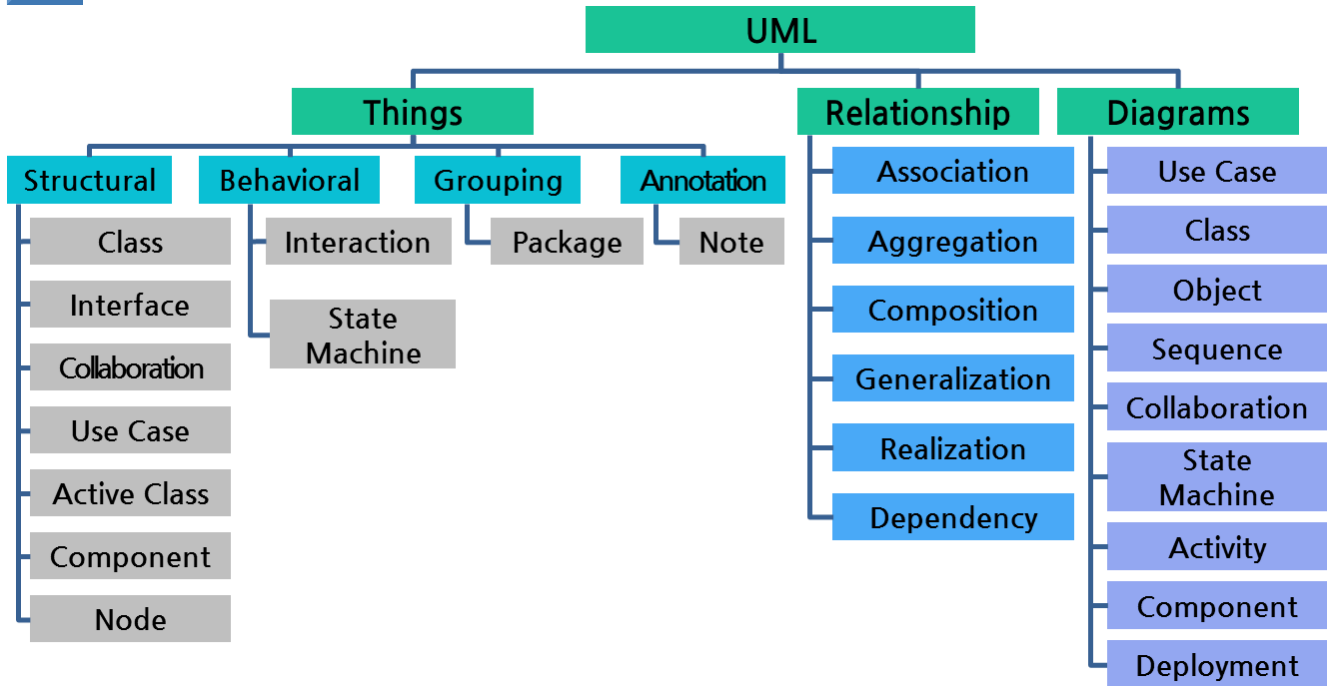
## 1 UML이란?

### 3 UML 특징

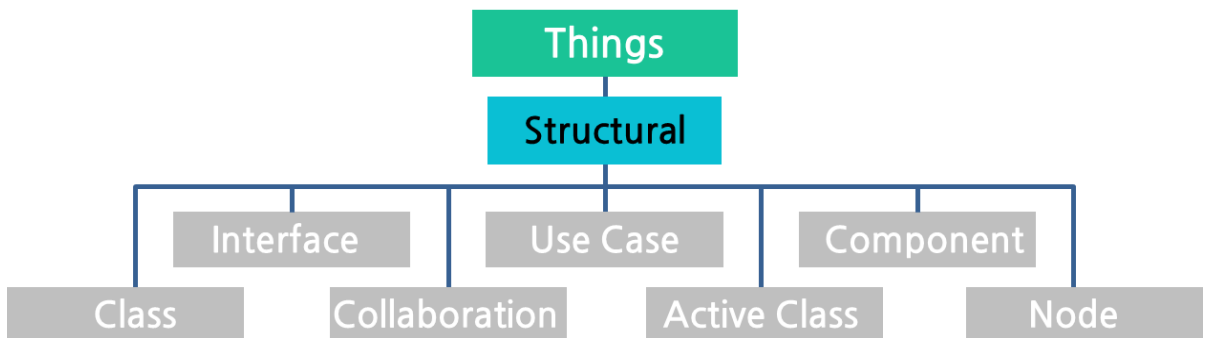


# UML 개요

## 2 UML 기본 구성



### 1 사물(Things)



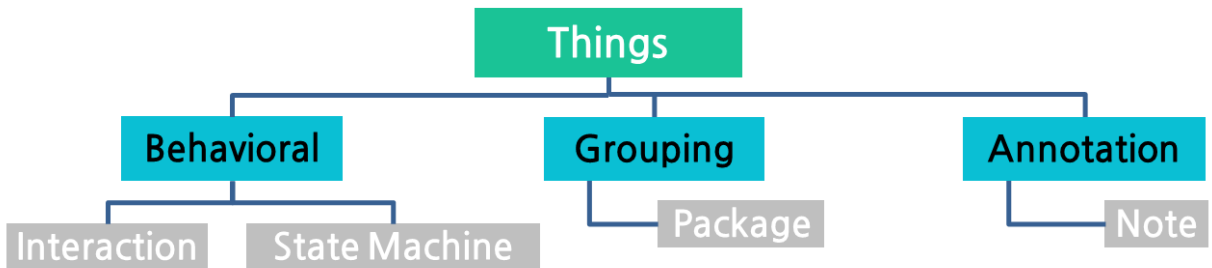
#### 구조사물(Structural Things)

UML 모델의 명사형, 모델의 정적인 부분, 개념적/물리적 요소를 표현

# UML 개요

## 2 UML 기본 구성

### 1 사물(Things)



#### 행동 사물 (Behavioral Things)

시스템의 행위

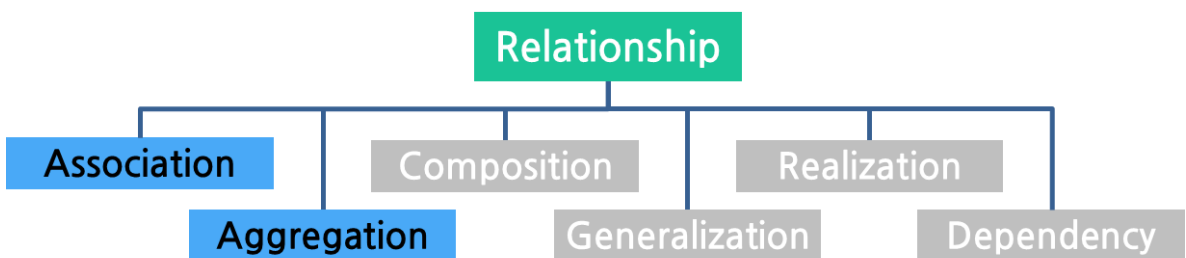
#### 그룹 사물 (Grouping Things)

개념을  
그룹화하는 사물

#### 주해 사물 (Annotation Things)

부가적으로 개념을  
설명하는 사물

### 2 관계(Relationship)



#### 연관 관계 (Association)

어느 한 사물이 다른 사물과  
연관이 있음을 표시하는 관계

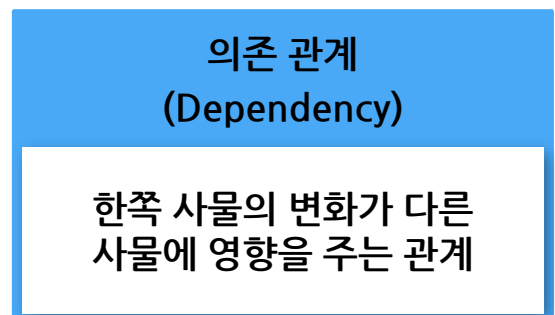
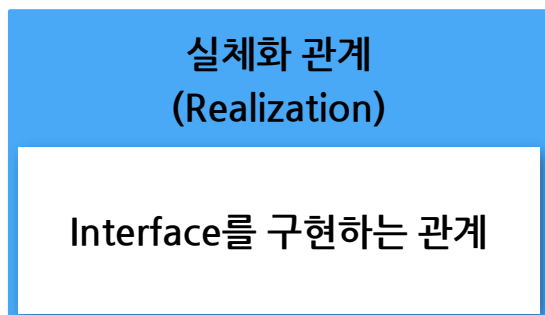
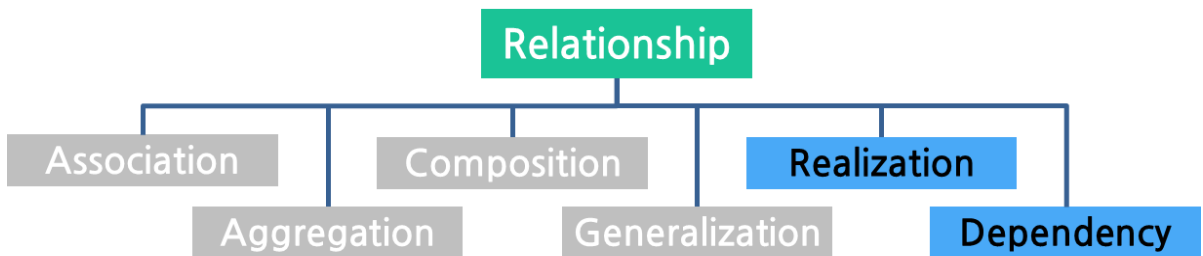
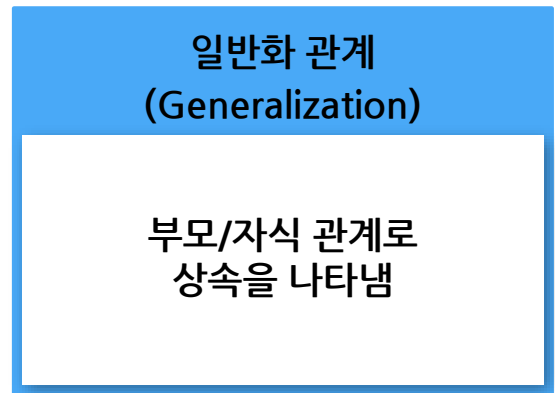
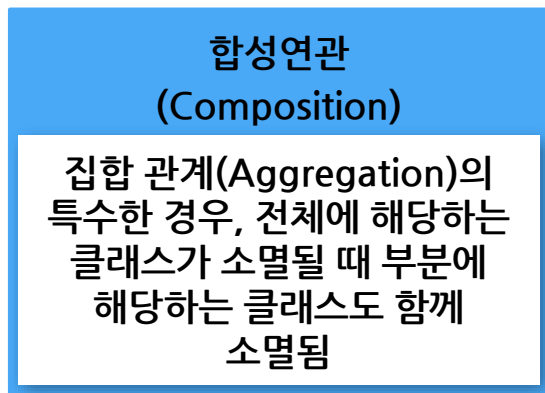
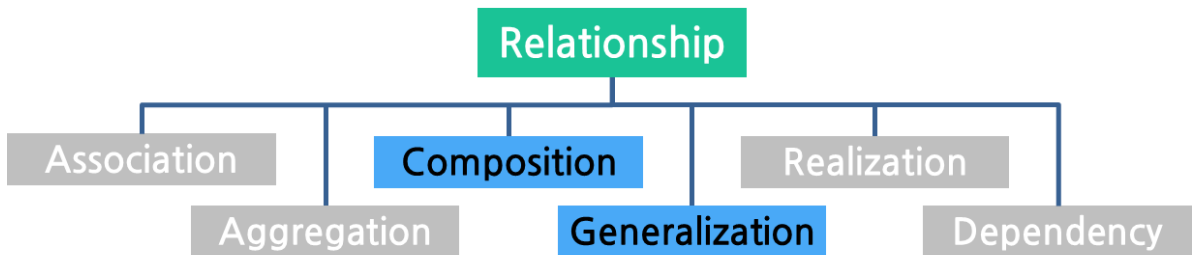
#### 집합 관계 (Aggregation)

전체와 부분 간의 관계

# UML 개요

## 2 UML 기본 구성

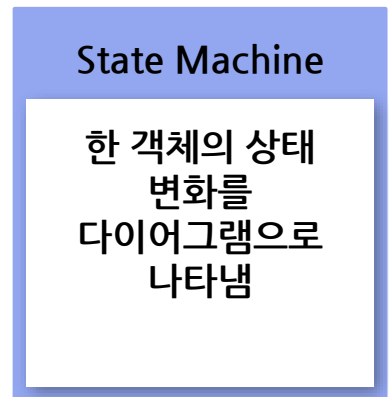
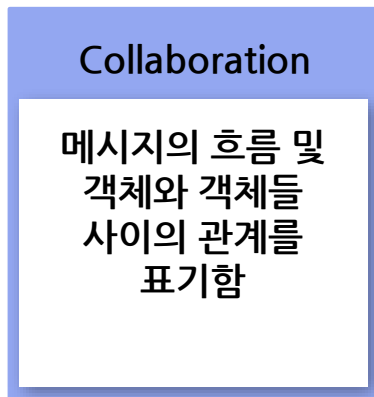
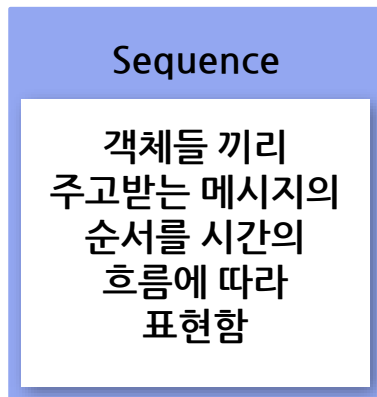
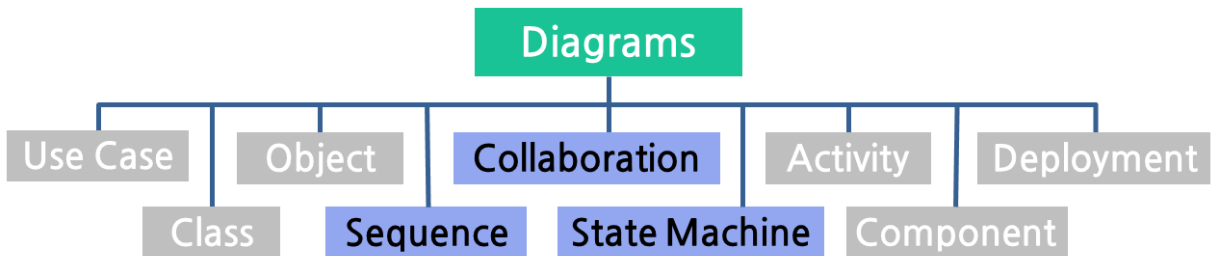
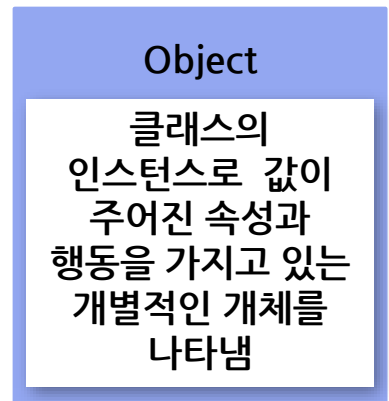
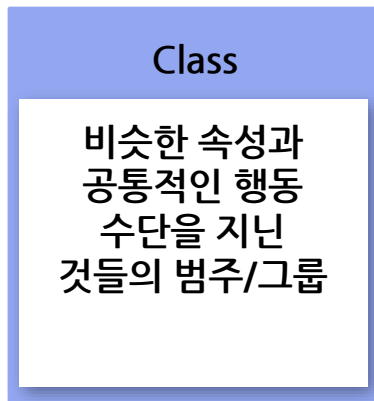
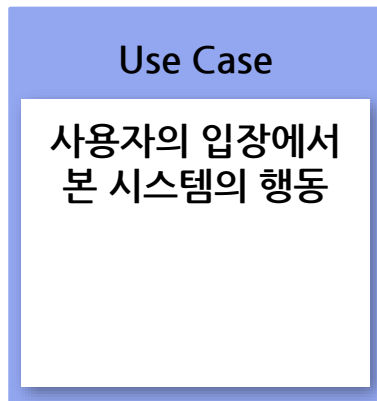
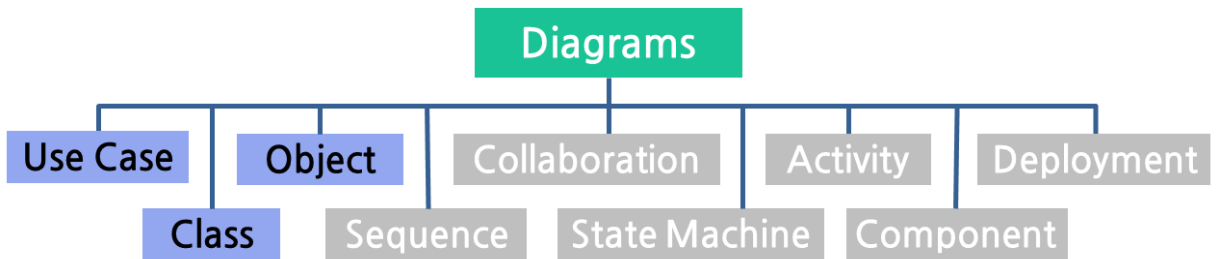
### 2 관계(Relationship)



# UML 개요

## 2 UML 기본 구성

### 3 도해(Diagrams)

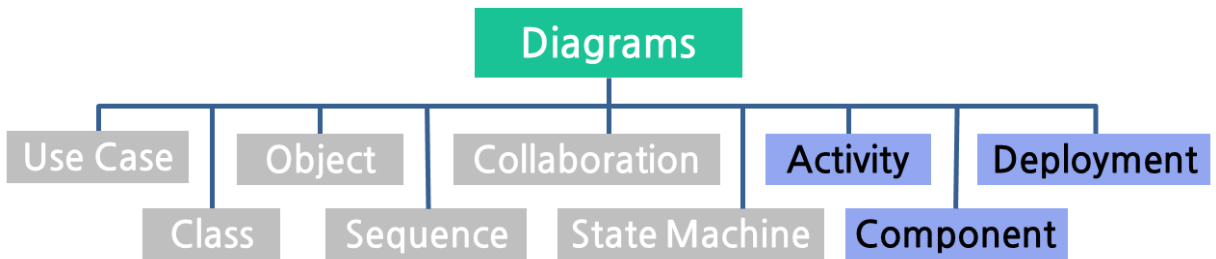




# UML 개요

## 2 UML 기본 구성

### 3 도해(Diagrams)



#### Active Class

Use Case 내부  
혹은 객체의 동작  
중에 발생하는  
활동을 표현함

#### Component

소프트웨어의  
물리적 단위(exe,  
dll 등 library)의  
구성과 연결상태를  
표현함

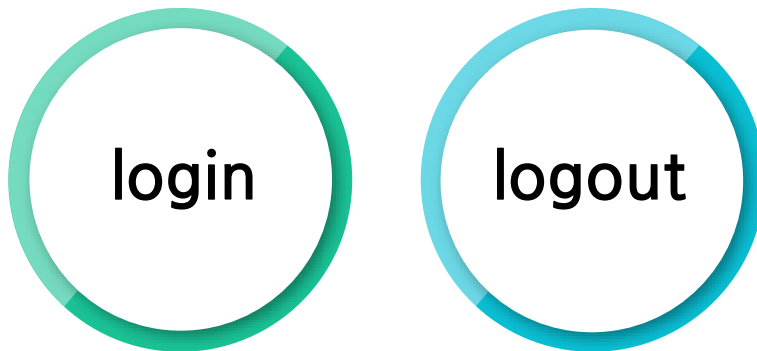
#### Deployment

실제 하드웨어적인  
배치와 연결상태를  
나타냄

# UML 개요

## 3 UML 표기법

### 1 Use Case



타원으로 표시하고 안쪽에 Use Case 명을 기술

‘~ 한다’와 같은 동사 형태

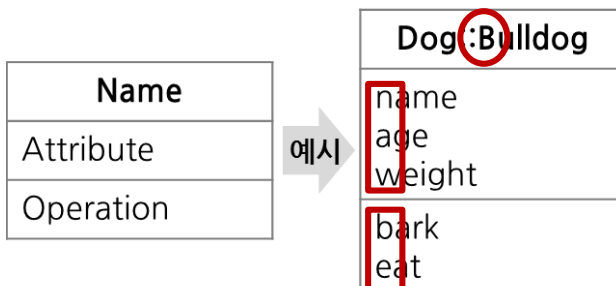
### 2 Class

#### 1 Class의 정의



Class

공통의 속성, 오퍼레이션 및 관계를 갖는 객체들의 집합



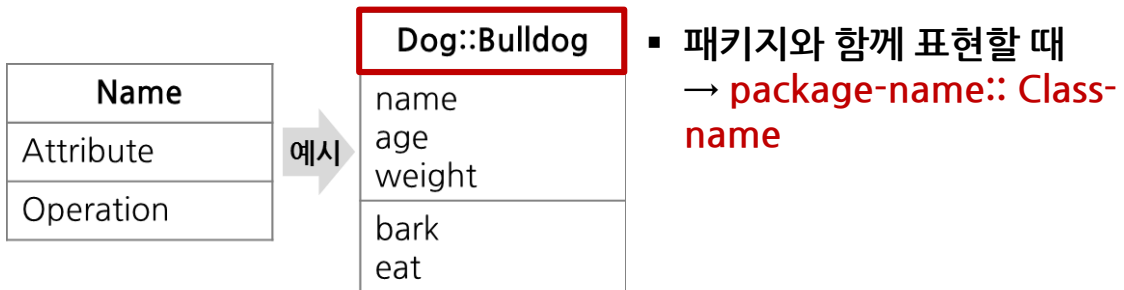
- 클래스의 이름은 **대문자**로 시작함
- 속성과 오퍼레이션은 **소문자**로 시작함

# UML 개요

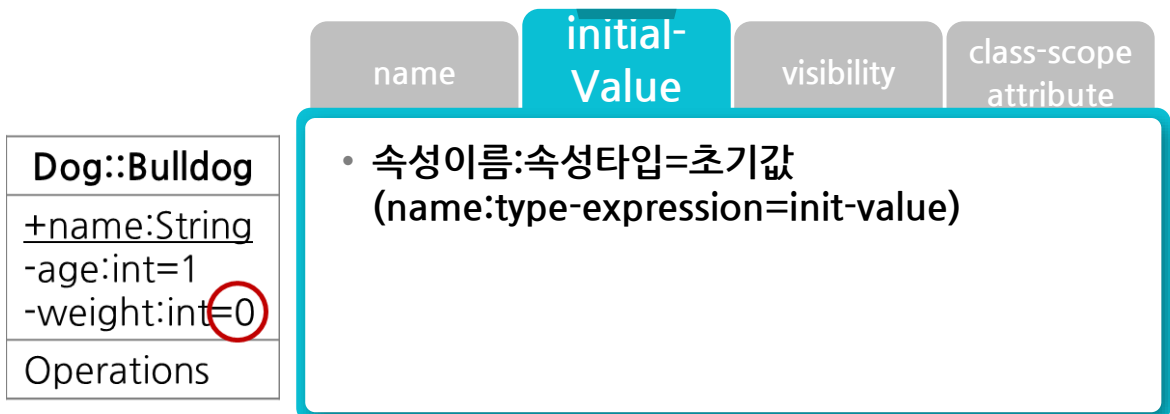
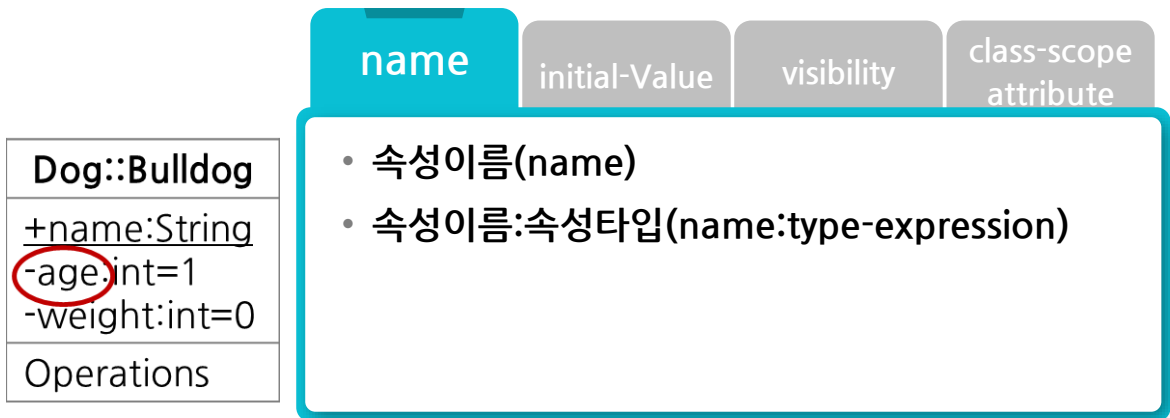
## 3 UML 표기법

### 2 Class

#### 1 Class의 정의



#### 2 속성(Attribute) 객체들의 특성을 표시

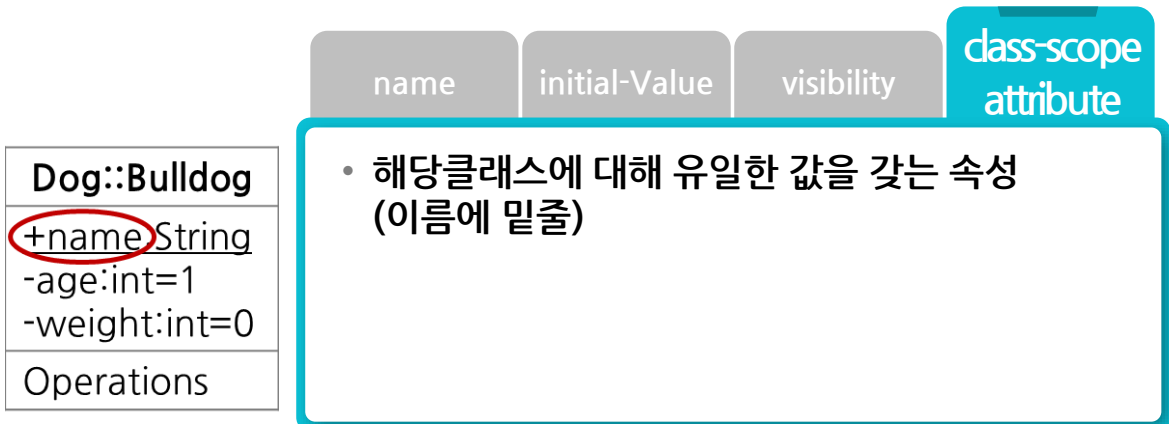
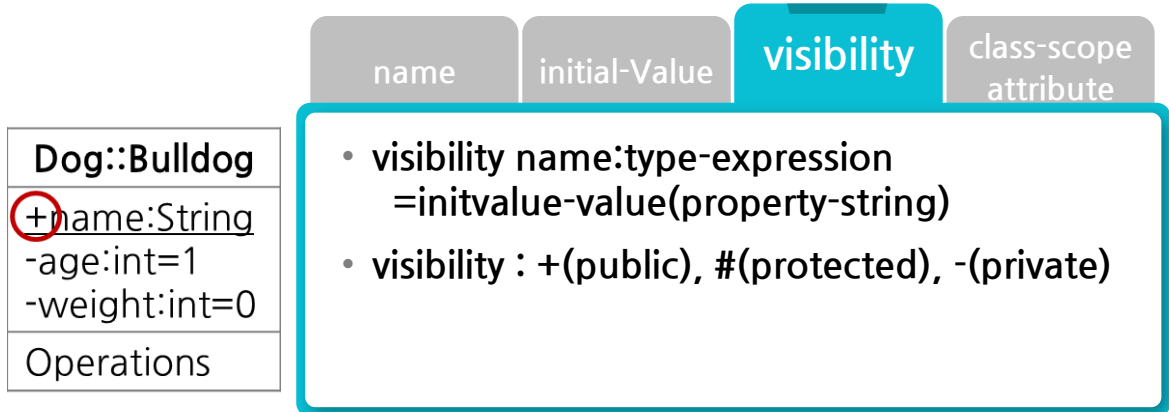


# UML 개요

## 3 UML 표기법

### 2 Class

#### 2 속성(Attribute) 객체들의 특성을 표시

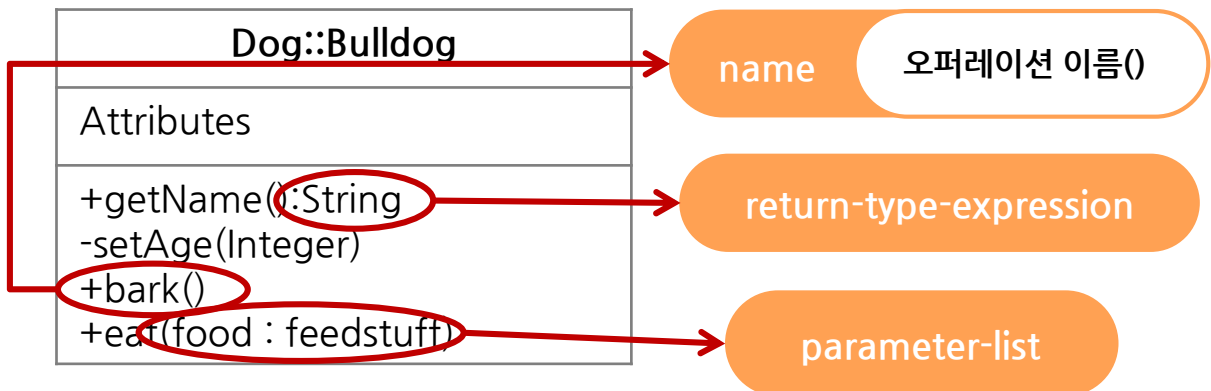
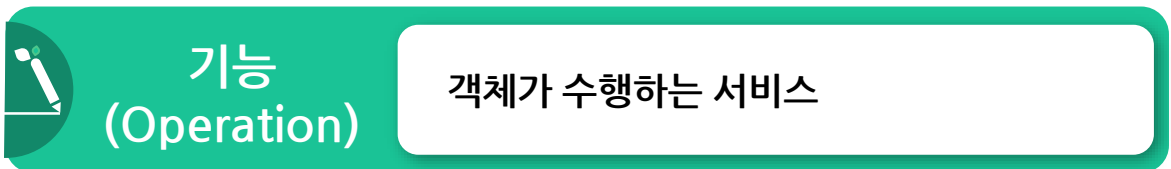


# UML 개요

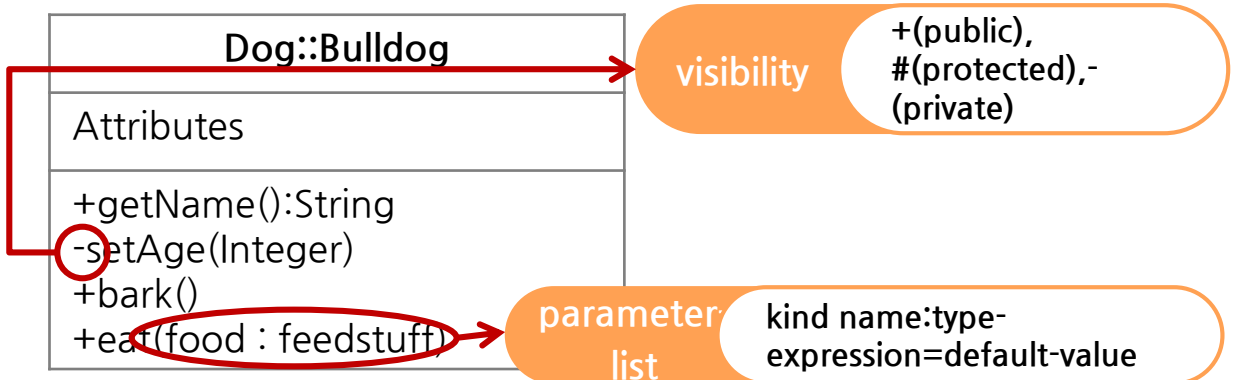
## 3 UML 표기법

### 2 Class

### 3 기능(Operation)



- 오퍼레이션 이름(파라미터 리스트) : 리턴타입
- visibility name(parameter-list):return-type-expression(property-string)



class-scope  
operation

해당 클래스에 대해 유일한 값을 갖는  
오퍼레이션(이름에 밀출)  
ex) static public String getName();

# UML 개요

## 3 UML 표기법

### 2 Class

#### 4 가시성(Visibility)

클래스의 속성과 오퍼레이션에 부여할 수 있는 특성으로  
다른 클래스에서 사용 가능 여부를 나타냄

구분	내용
+(public)	외부 클래스에서도 사용 가능함을 의미
#(protected)	하위 클래스와 해당 클래스 내에서만 사용 가능함을 의미
-(private)	해당 클래스에서만 사용 가능함을 의미
~(package, default)	패키지 내에서만 사용 가능함을 의미

#### 5 추상클래스(Abstract Class)

추상메소드가 1개 이상 존재하는 클래스

내부 구현 코드가 정의되어 있지 않은 메소드

Name(Abstract Class)
Attributes
+abstractMethod

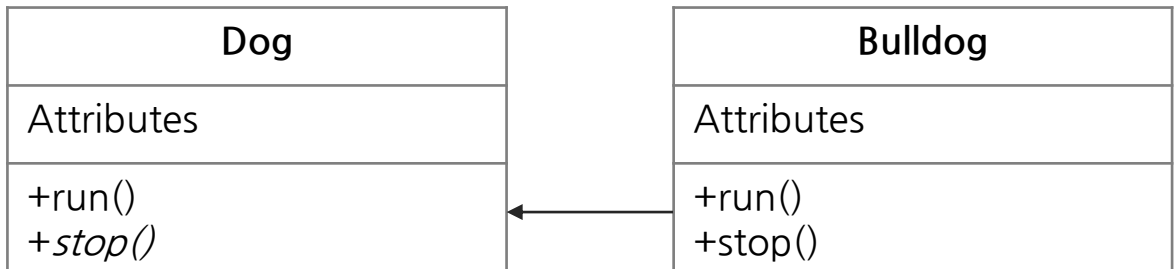
(※ 추상클래스와 메소드는 **이탤릭체**로 표기)

# UML 개요

## 3 UML 표기법

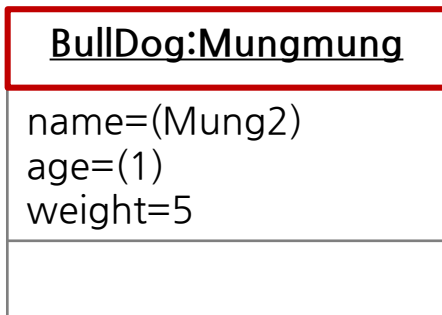
### 2 Class

#### 5 추상클래스(Abstract Class)



(※ 추상클래스와 메소드는 **이탤릭체**로 표기)

### 3 객체(Object)



#### 객체(Object)

- Class의 Instance
- 특정한 속성 값을 가짐
- Object 이름은 **밑줄로 표현**  
ex) Class-name : Object-name

# UML 개요



## 3 UML 표기법

### 4 협력(Collaboration)

구현관점에서 목적을 달성하기 위한 일련의 행위

Collaboration

세차하기

- 타원을 **점선으로 표기**
- 타원 안에 **역할의 내용**을 기입

### 5 상태머신(State Machine)

객체의 상태를 순서대로 명시한 것

response

- 이벤트에 대한 객체의 **응답**과 **반응**

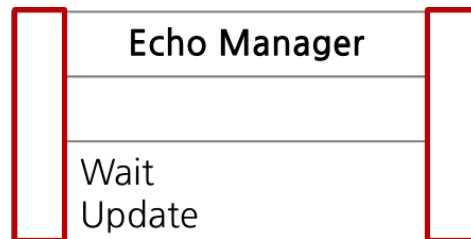


## UML 개요

### 3 UML 표기법

#### 6 Active Class

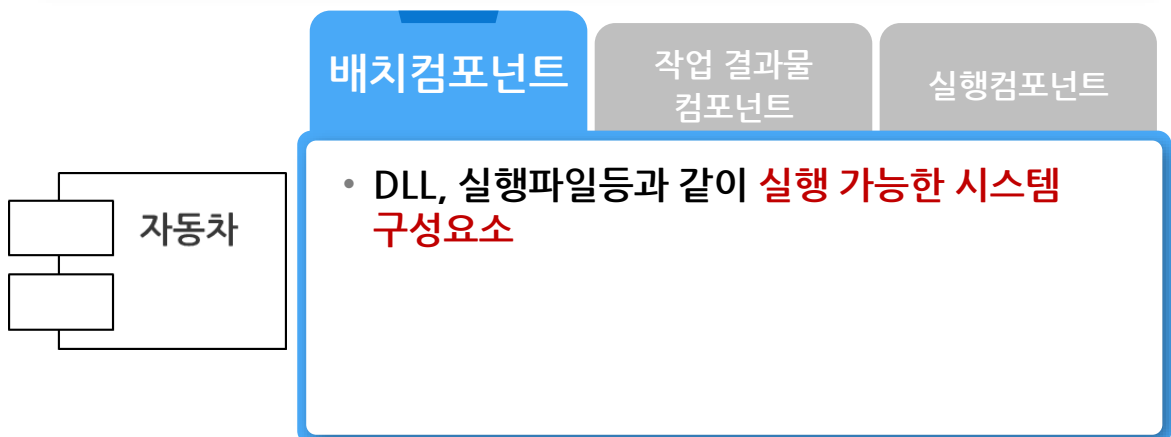
하나 이상의 프로세스나 쓰레드를 갖는 객체를 파생하는 클래스 기술



- Class 표기와 비슷하지만, 양 옆에 세로라인이 추가됨

#### 7 Componen

시스템을 구성하는 물리적인 단위로 독립적인 실행단위 혹은 배포단위



# UML 개요

## 3 UML 표기법

### 7 Component



배치컴포넌트

작업 결과물  
컴포넌트

실행컴포넌트

- 시스템의 실행에 직접적으로 참여하지는 않지만, 개발 작업에서 만들어지는 결과들이며 **실행 시스템을 생성**하는데 사용됨
- 분석/ 설계 문서, 소스코드 파일, 데이터 파일 등이 있음



배치컴포넌트

작업 결과물  
컴포넌트

실행컴포넌트

- 실행 시스템의 수행결과로 생성되는 컴포넌트
- 메모리상에 존재하는 객체
- 실행 결과로 생성된 DB 레코드 또는 파일

# UML 개요

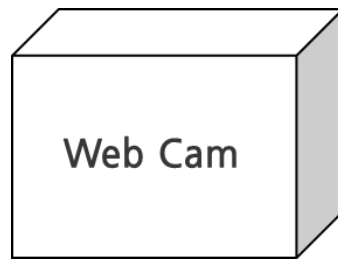


## 3 UML 표기법

### 8 Node

물리적인 요소로서 시스템이 실행 될 때 존재함

어느 정도의 메모리와 처리 능력을 갖는 전산 자원을 의미



- **육면체로 표현**하며 이름을 표기하고 필요에 따라 탑재되는 컴포넌트를 표기함

# UML 실습



## 1 StartUML 다운로드 및 설치

1 항목

2 설치

## 2 화면 설명

## 3 간단하게 그리는 방법 설명

### ▪ 활용 예제-수강신청 예제

#### 수강신청 요구 사항

- 수강신청 담당자는 시스템에 접속하여 수강신청 항목을 입력한다.
- 학생들은 수강신청 시스템에 접속하여 원하는 강좌를 조회, 선택, 신청을 한다.
- 수강 신청이 완료되면 수강 신청 담당자는 해당 강좌의 교수에게 수강신청 명단을 전달한다.

- 1) Use case Diagram 그리기
- 2) Sequence Diagram 그리기
- 3) Collaboration Diagram 그리기
- 4) Activity Diagram 그리기
- 5) Statechart Diagram 그리기

## 학습정리

### 1. UML 개요



- 업무, 소프트웨어 애플리케이션, 시스템 아키텍처를 모델링 하는데 사용되는 표준 비주얼 모델링 언어
- UML은 사물(Things), Relationship(관계), Diagram(도해)으로 구성
- UML은 가시화 언어, 명세화 언어, 구축언어, 문서화 언어로서의 특징을 가짐

## 학습정리

### 2. UML 실습



- UML은 한 회사에서 독점하고 있지 않고 모두에게 개방되어 있음
- Use Case Diagram은 사용자의 입장에서 본 시스템의 행동을 나타냄
- Class Diagram은 비슷한 속성과 공통적인 행동 수단을 지닌 것들의 그룹을 나타냄
- Sequence Diagram은 객체들끼리 주고받는 메시지의 순서를 시간의 흐름에 따라 나타냄
- Active Class Diagram은 Use case 내부 혹은 객체의 동작 중에 발생하는 활동을 표현함