

# Introduction to UML

201111339 김민우  
201111344 김재엽  
201211386 최하나

# 목차

- ▶ 1 UML 이란
- ▶ 2 UML 특징
- ▶ 3 UML 의 구성요소
  - 3.1 UML Building Blocks 구성요소
    - 사물(Things)
    - 관계(Relationship)
    - 다이어그램(Diagram)
- ▶ 4 UML 모델링 Tools : CASE

# 1 UML 이란

- ▶ UML(Unified Modeling Language)  
: 객체 지향 분석/설계용의 모델링 언어
- ▶ UML은 Booch, Rumbaugh, Jacobson이 주장하는 각각의 객체지향방법론 중에서 장점들을 통합하여 여러 가지 방법론들을 모두 표현할 수 있도록 만든 언어이다.
- ▶ UML은 소프트웨어 개발에 사용되는 다이어그램을 정의하는 것으로, 소프트웨어 개발 시 산출물들을 시각적으로 제공함으로써 상호 간의 의사 소통을 원활하게 할 수 있다.

# 2 UML 특징

## ▶ 가시화 언어

소프트웨어의 개념 모델을 시각적인 형태로 작성하고, Symbol에 명확한 정의가 존재하므로 개발에 있어 원활한 의사소통이 가능하다.

## ▶ 명세화 언어

명세화란 정확하고 명백하며, 완전한 모델을 만드는 것을 의미한다.

UML은 소프트웨어 개발과정인 분석/설계/구현 단계의 각 과정에서 필요한 모델을 명세화 할 수 있는 언어이다.

## ▶ 구축 언어

UML로 명세화된 설계모델은 JAVA, C++, VB등 다양한 언어의 소스코드로 변환하여 구축할 수 있다.

반대로 구축되어있는 소스코드를 UML로 변환하여 분석하는 Reverse도 가능하다.

## ▶ 문서화 언어

시스템 아키텍처와 이에 대한 모든 상세 내역에 대한 문서화를 다루며, 요구사항을 표현하고 시스템을 테스트 하는 언어도 제공한다.

# 3 UML 의 구성요소

- ▶ **UML building blocks**  
( UML을 구성하는 요소들 즉 사물(Things), 관계(Relationship), 다이어그램(Diagram) )
- ▶ **Rules to connect the building blocks**  
( UML을 구성하는 요소들을 연결하는 규칙 )
- ▶ **Common mechanisms of UML**  
( UML의 메커니즘 )

# 3.1 UML Building Blocks 구성요소

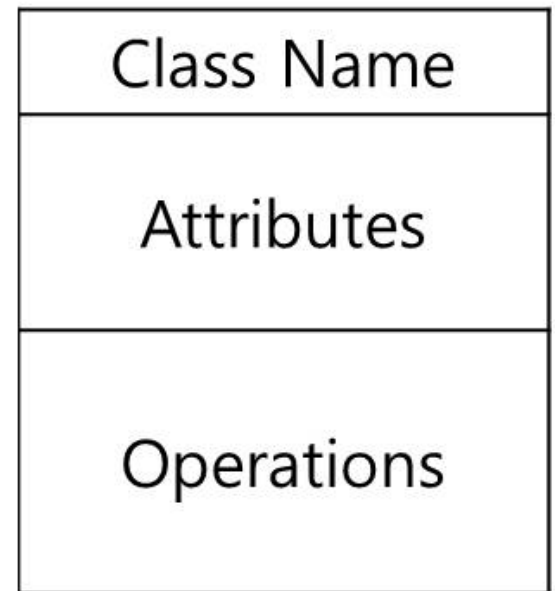
- ▶ **사물(Things)** : 추상적인 개념으로 UML에서는 4가지 사물로 나뉜다.
  - 구조사물(Structural):모델의 정적인 부분들을 정의. 물리적, 개념적 요소를 표현
    - 클래스, 인터페이스, 통신, 유스케이스, 활성클래스, 컴포넌트, 노드 의 7개의 구조사물과 7개 기본 구조사물들의 변이(Variation)가 있다
    - 클래스의 변이로는 액터, 신호(Signal), 유틸리티(Utility)  
컴포넌트의 변이로는 애플리케이션, 문서, 파일, 라이브러리 등
  - 행동사물(Behavioral):UML모델의 동적인 부분으로 구성
    - 교류 와 상태 머신이 있다.
  - 그룹사물 (Grouping) : UML모델의 요소들을 그룹화
    - 패키지가 있다.
  - 주해사물 (Annotational) : UML모델을 설명 (주석)
    - 노트가 있다.

# 구조사물 (Structural)

## ▶ 클래스 (Class)

공통의 속성들, 오퍼레이션들, 관계들, 의미들을 공유하는 개체들의 집합에 대한 기술 (description) 을 뜻한다.

직사각형으로 표기하며  
이름(Class Name),  
속성(Attributes),  
오퍼레이션(Operations) 을 넣는다.



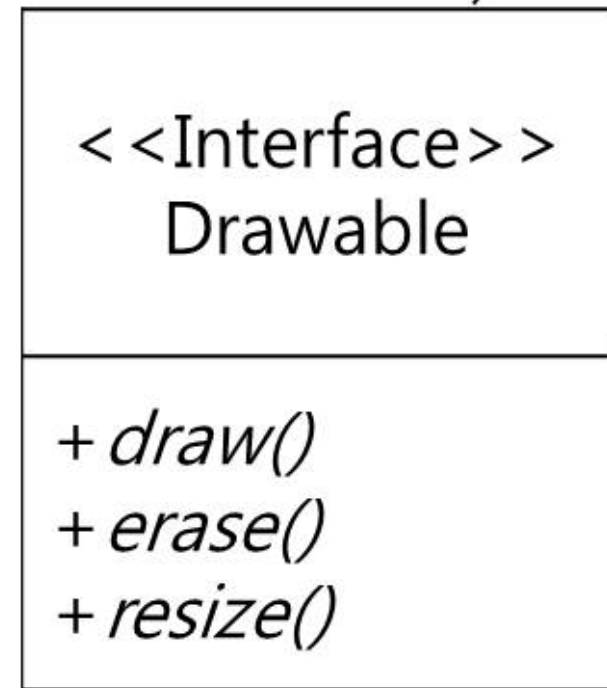
# 구조사물 (Structural)

## ▶ 인터페이스 (Interface)

역할을 표현하는 UML 모델 요소로서 객체들에 의해 실현된다.

역할이 제공해야 하는 서비스들은 속성과 오퍼레이션으로 작성된다.

인터페이스는 <<interface>> 키워드를 갖는 클래스 기호로 표현한다.





# 구조사물 (Structural)

## ▶ 통신 (Communication)

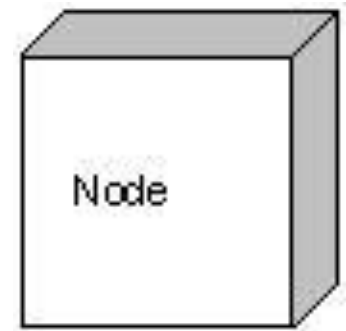
교류를 정의하며, 서로 다른 요소화 역할들이 모여 있다.

통신은 행동적이고 구조적인 중요성을 가진다.

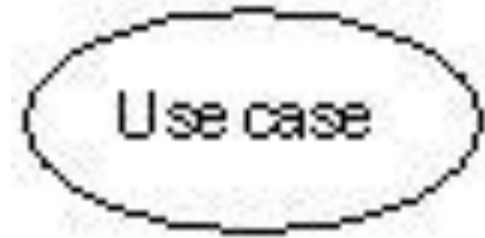


## ▶ 노드 (Node)

런타임 중에 존재하는 물리적인 요소



# 구조사물 (Structural)



## ▶ 유스케이스 (Use Case)

유스케이스란 개발될 시스템의 개개 액터가 시스템 사용 목적을 잘 달성할 수 있도록 개발될 시스템이 제공해야 하는 서비스이다.

(액터란 시스템과 상호 작용해야 하는 어떤 사람 또는 어떤 것이라고 정의한다.)

UML에서는 사용자의 목적 달성을 위해 시스템이 제공해야 하는 서비스와 서비스를 제공하기 위한 과정을 유스케이스로 표현한다.

# 구조사물 (Structural)

- ▶ **활성 클래스 (Active Class)**

객체가 하나 또는 그 이상의 프로세스(process)나 스레드(Thread)를 갖는 클래스

클래스 이지만 그 객체들의 행동이 다른 요소들과 함께 동시적으로 이루어진다는 점이 다르다.

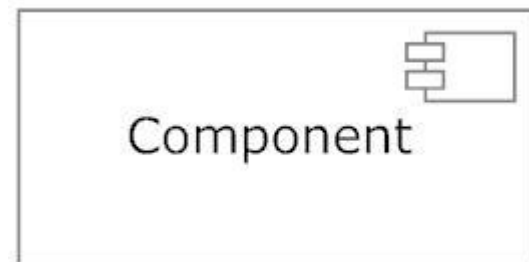


# 구조사물 (Structural)

## ▶ 컴포넌트 (Component)

컴포넌트는 내부를 캡슐화하고 컴포넌트의 외부 환경 안에서 전체적으로 대체 가능한 시스템의 모듈이다.

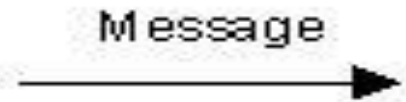
컴포넌트기반개발의 가장 큰 장점은 재사용이다. 기존의 부품을 재사용함으로써 개발기간 및 개발비용을 줄일 수 있게 된다.



# 행동사물 (Behavioral)

## ▶ 교류 (Interaction)

객체들간에 주고받는 메시지로 구성된다.  
직선으로 나타내며, 항상 오퍼레이션  
이름을 포함한다.



## ▶ 상태 머신 (State Machine)

개별 클래스의 행동이나 여러 클래스의 특정 통신  
의 행동을 하나의 머신으로 지정할 수 있다.

상태 머신은 많은 요소를 갖는데  
상태전이, 사건, 활동이 있다.



# 그룹사물 (Grouping)

## ▶ 패키지 (Package)

UML에서는 구성 요소 그룹을 패키지로 표현하고, 패키지에 포함되는 구성 요소들은 멤버라고 부른다.

패키지는 개념적이다.

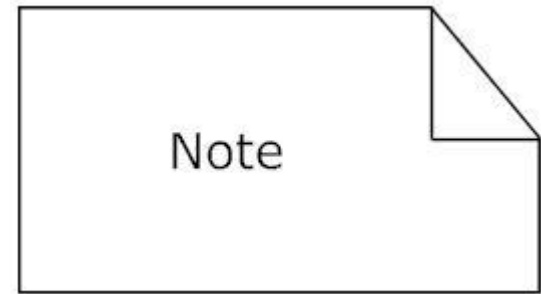
패키지의 종류로는

프레임워크, 모델, 서브시스템과 같은 변이가 있다.



# 주해사물 (Annotational)

- ▶ 노트 (Note)  
주석을 다는 용도로 사용된다.



## 3.1 UML Building Blocks 구성요소

- ▶ **관계(Relationship)** : 요소가 서로 연관되는 방법을 보여준다.
  - 의존 (Dependency) : 두 사물간의 의미적 관계, 하나의 요소의 변화는 다른 하나에 영향을 미치는 두 가지 사이의 관계.  
점선으로 된 직선을 사용하며, 의존하고 있는 사물을 향하고 있다.

Dependency





# 3.1 UML Building Blocks 구성요소

- 연관 (Association) : 구조적 관계로서 어느 한 사물 객체가 다른 사물 객체와 연결되어 있음을 말한다.



- 일반화 (Generalization) : 클래스와 인터페이스 사이에서 상속관계를 설명한다.



- 실체화 (realization) : 인터페이스와 클래스나 컴포넌트 사이의 관계를 지정한다.



# 3.1 UML Building Blocks 구성요소

## ▶ 다이어그램(Diagram)

- 다이어그램은 요소들을 그림으로 표현한 것이다. 시스템을 가시화하기 위하여 사용한다.
- UML의 다이어그램은 두가지 유형의 다이어그램이 있다.

구성요소를 표현하기 위한 구조적 다이어그램  
(Structural UML Diagrams)

행위를 표현하기 위한 행위 다이어그램  
(Behavioral UML Diagrams)

# 구조적 다이어그램 (Structural UML Diagrams)

- ▶ 클래스 다이어그램 ( Class diagram )
- ▶ 객체 다이어그램 ( Object diagram )
- ▶ 패키지 다이어그램 ( Package diagram )
- ▶ 컴포넌트 다이어그램 ( Component diagram )
- ▶ 복합 구조 다이어그램  
( Composite structure diagram )
- ▶ 배치 다이어그램 ( Deployment diagram )
- ▶ 프로필 다이어그램 ( Profile diagram )

# 구조적 다이어그램 (Structural UML Diagrams)

- ▶ **클래스 다이어그램 ( Class diagram )**

클래스, 인터페이스, 통신 간의 관계를 나타내며 가장 공통적으로 쓰이는 다이어그램이다.

클래스(Class) 와 인터페이스(Interface), 열거형(Enumeration), 패키지(Package) 및 여러 가지 관계들 뿐만 아니라 인스턴스(Instance)와 그것들의 연결(Link) 등도 포함한다.

# 구조적 다이어그램 (Structural UML Diagrams)

- ▶ **객체 다이어그램 ( Object diagram )**

시스템을 구성하는 객체, 객체간의 관계를 표현하는 다이어그램이다.

특정 시점의 객체들의 구조적 상태를 표현한다.

- ▶ **패키지 다이어그램 ( Package diagram )**

UML의 다양한 모델요소를 그룹화시킨 다이어그램이다.

# 구조적 다이어그램 (Structural UML Diagrams)

- ▶ **컴포넌트 다이어그램 ( Component diagram )**  
시스템을 구성하는 컴포넌트와 컴포넌트가 제공하거나 이용하는 인터페이스를 표현한 다이어그램

소프트웨어의 시스템을 구성하는 주요 기능적 요소들을 표현할 때 사용한다.

소프트웨어 컴포넌트 사이의 의존관계를 묘사한다.  
소프트웨어 컴포넌트를 구성하는 요소들과 그것들을 구현하는 요소들도 모두 표현될 수 있다.

# 구조적 다이어그램 (Structural UML Diagrams)

## ▶ 복합 구조 다이어그램

### ( Composite structure diagram )

컴포넌트의 내부 구조를 표현하는데 사용  
파트, 포트, 연결자 등의 개념을 이용하여 컴포넌트  
가 이용하는 인터페이스와 내부 구성요소 간의 관  
계를 표현하는 다이어그램

분류자(Classifier)의 내부 구조를 표현한다. 여기  
에 분류자가 시스템의 다른 부분들과의 상호작용  
하는 지점 등을 포함한다.

# 구조적 다이어그램 (Structural UML Diagrams)

- ▶ **배치 다이어그램 ( Deployment diagram )**  
시스템을 구성하는 물리적인 노드(Node)와 통신 경로, 컴포넌트의 수행환경을 표현하는 다이어그램
- ▶ **프로필 다이어그램 ( Profile diagram )**  
<< profile >> 스테레오 타입 패키지로 스테레오 타입 클래스, 프로파일 등의 고정 관념을 표시하는 다이어그램



# 행위 다이어그램 (Behavioral UML Diagrams)

- ▶ 유스케이스 다이어그램 ( Use case diagram )
- ▶ 활동 다이어그램 ( Activity diagram )
- ▶ 상태 다이어그램 ( State diagram )
- ▶ 시퀀스(순차) 다이어그램 ( Sequence diagram )
- ▶ 통신 다이어그램 ( Communication diagram )
- ▶ 타이밍 다이어그램 ( Timing diagram )
- ▶ 상호작용 개요 다이어그램  
( Interaction overview diagram )

# 행위 다이어그램 (Behavioral UML Diagrams)

- ▶ **유스케이스 다이어그램 ( Use case diagram )**  
시스템의 기능적 요구사항을 표현하고, 입출력을 하는 대상을 액터로 정의, 시스템이 제공할 기능을 유스케이스로 표현하는 다이어그램

특정 시스템 혹은 개체내의 유스케이스(Use Case)들과 그 외부의 액터(Actor)들 간의 관계를 표현한 것이다.

유스케이스는 해당 시스템의 기능을 표현하며 그것들이 어떤 외부 액터들과 상호작용하는지를 나타낸다.

# 행위 다이어그램 (Behavioral UML Diagrams)

## ▶ 활동 다이어그램 ( Activity diagram )

여러 활동들이 순차, 병행 방식등을 수행하는 상황을 표현한 다이어그램

활동들의 수행 흐름을 묘사하는데 적합하다. 일반적으로 작업흐름(Workflow)을 표현하기 위해 많이 사용되며, 클래스, 패키지 혹은 연산 등의 개체에 대해 주로 사용된다.

# 행위 다이어그램 (Behavioral UML Diagrams)

- ▶ **상태 다이어그램 ( State diagram )**

시스템 구성 요소의 동적인 행위를 표현한 다이어그램

특정 개체의 동적인 행위를 상태(State)와 그것들 간의 전이(Transition)를 통해 나타낸다. 일반적으로 클래스의 인스턴스에 대한 행위를 나타내는데 사용한다.

# 행위 다이어그램 (Behavioral UML Diagrams)

- ▶ 시퀀스(순차) 다이어그램 ( Sequence diagram )  
시간적 순서에 따른 상호작용을 표현한 다이어그램

인스턴스들이 어떻게 상호작용을 하는지를 나타낸다. 인스턴스(Instance)들 상호간에 주고받는 자극(Stimulus)들을 직접적으로 표현한다.

# 행위 다이어그램 (Behavioral UML Diagrams)

- ▶ **통신 다이어그램 ( Communication diagram )**  
상호작용에 참여하는 생명선(객체, 컴포넌트)간의 관계를 명시적으로 표현한 다이어그램
- ▶ **타이밍 다이어그램 ( Timing diagram )**  
상호작용에 참여하는 생명선의 상태 변화, 메시지에 대하여 정확한 시간적 정보를 표현한 다이어그램
- ▶ **상호작용 개요 다이어그램  
( Interaction overview diagram )**  
상위 수준의 상호작용인 시퀀스, 통신, 타이밍 다이어그램 등으로 표현된 여러 개의 상호작용간의 관계를 표현한 다이어그램

# 4 UML 모델링 Tools : CASE

- ▶ UML과 같은 모델링 언어를 이용하여 모델링을 하는 도구(Tool)를 CASE(Computer Aided Software Engineering) 도구라 한다.
- ▶ CASE 도구는 상용, 무료 오픈소스, Eclipse 플러그인 형태까지 다양하게 존재한다.
- ▶ IBM사의 ( Rational Rose ), Borland사의 ( Together ), 무료 오픈 소스기반의 ( StarUML ), 외에 ( Visual Paradigm ), MS ( Visio ) 등이 있다.

# 4 UML 모델링 Tool : StarUML

## ▶ 오픈 소스 소프트웨어 모델링 도구

### ▶ 다양한 언어를 지원한다.

- Java , C++ , C#

### ▶ Design Pattern을 지원한다.

- GoF, EJB Patterns

### ▶ StarUML User Guide

- [http://staruml.sourceforge.net/docs/user-guide\(ko\)/toc.html](http://staruml.sourceforge.net/docs/user-guide(ko)/toc.html)