

소프트웨어공학 활용



다이어그램 및 프로세스 모델링 도구



한국기술교육대학교
온라인평생교육원

학습내용

- 다이어그램 도구
- 프로세스 모델링 도구

학습목표

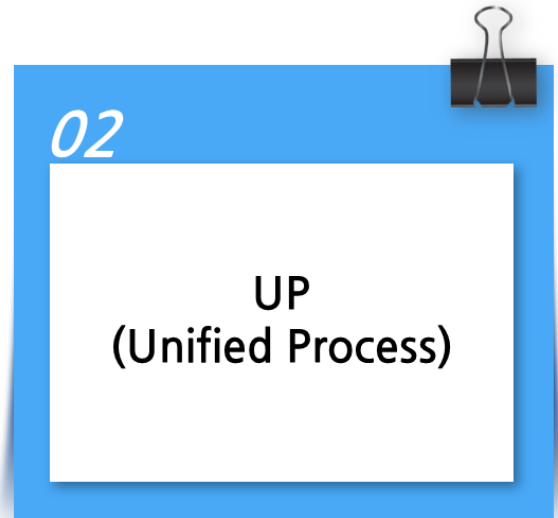
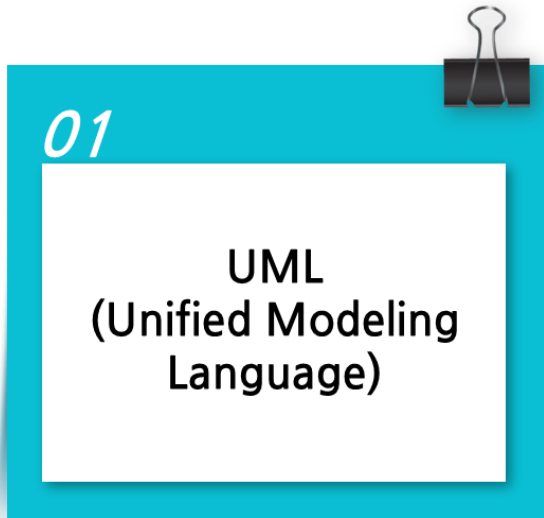
- CASE 도구 다이어그램 도구를 이해하고 설명할 수 있다.
- CASE 도구 프로세스 모델링 도구를 이해하고 설명할 수 있다.

다이어그램 도구

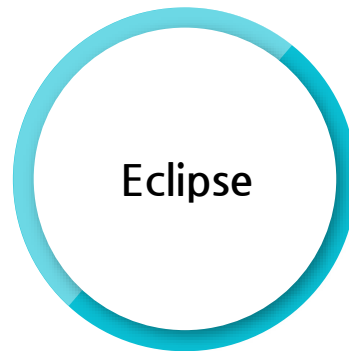


1 CASE 도구

1 모델링 관련 도구



2 통합 개발 환경(IDE)



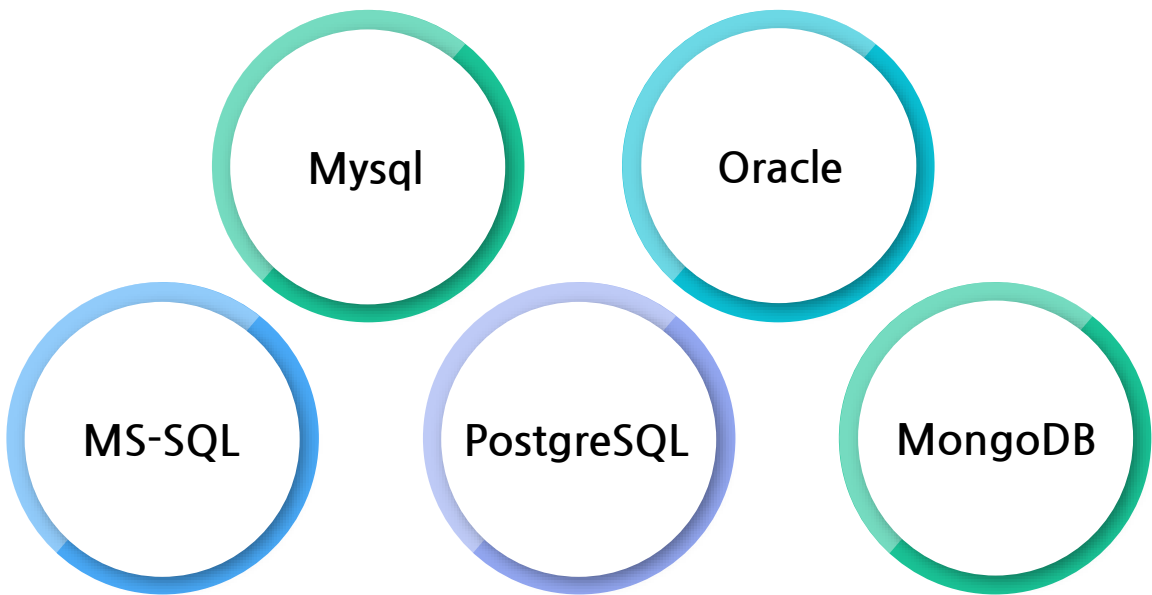
프로그램 개발 툴

다이어그램 도구



1 CASE 도구

3 DBMS



4 UML 모델링 도구 선택 기준

1 제품의 상태

버전업 여부, 오픈소스의 경우 해당 프로젝트의 활성 상태

2 Reverse / Forward engineering 지원 여부

3 UML 표준의 최신 스펙 지원 여부

4 사용 편의성과 시스템 요구 사항

5 플랫폼 지원 여부

다이어그램 도구



1 CASE 도구


5 CASE 도구(UML) 목록

이름 (Name)	적용 플랫폼 (Platform/OS)	라이선스 (Software License)	지원 언어 (Programming Language Used)
Modelio	Modeliosoft	Windows, Linux, macOS	Java
Papyrus	Commissariat à l'Énergie Atomique, Atos Origin	Windows, Linux, macOS (Java)	Java
UMLet	The UMLet Team	Windows, macOS, Linux	Java
Umbrello UML Modeller	Unix-like; Windows	GPL	C++, KDE
UML Designer	Windows, macOS, Linux	EPL	Java, Sirius

다이어그램 도구

2 Class Diagram

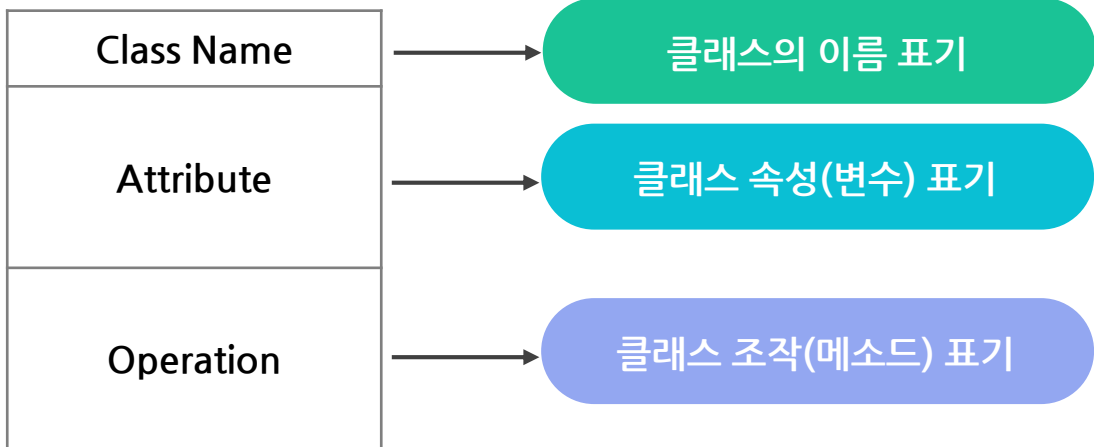
1 Class Diagram이란?



Class Diagram

- 클래스와 그들 간의 관계를 정의한 Diagram
- 정적인 관점에서 클래스 구조를 표현
- 데이터 모델링을 수행하는 용도도 포함됨
- 코딩을 위한 직접적인 설계 사양을 제공
- 논리적, 물리적인 관점까지 일관된 형식으로 시스템을 분석, 설계하는 도구로 사용

2 클래스 표기 형식



3 가시성(Visibility)

1 역할

속성(변수)이나 조작(메소드)을 외부에서 보는 법을 표현, 확인

클래스의 캡슐화 레벨을 표현, 확인

다이어그램 도구

2 Class Diagram

3 가시성(Visibility)

2 종류

Class Name
+ Attribute(public)
#Attribute(protected)
-Attribute(private)
~Attribute(package)
+Methods(public)
#Methods(protected)
-Methods(private)
~Methods(package)

+(public)

#(protected)

-(private)

~(package)

- 어떤 클래스에서든 접근 가능

Class Name
+ Attribute(public)
#Attribute(protected)
-Attribute(private)
~Attribute(package)
+Methods(public)
#Methods(protected)
-Methods(private)
~Methods(package)

+(public)

#(protected)

-(private)

~(package)

- 해당 클래스 내부와 서브 클래스로부터 접근 가능

다이어그램 도구

2 Class Diagram

3 가시성(Visibility)

2 종류

Class Name
+ Attribute(public)
#Attribute(protected)
-Attribute(private)
~Attribute(package)
+Methods(public)
#Methods(protected)
-Methods(private)
~Methods(package)

+(public)

#(protected)

-(private)

~(package)

- 해당 클래스 내부로부터만 접근 가능

Class Name
+ Attribute(public)
#Attribute(protected)
-Attribute(private)
~Attribute(package)
+Methods(public)
#Methods(protected)
-Methods(private)
~Methods(package)

+(public)

#(protected)

-(private)

~(package)

- 해당 클래스와 내부와 동일한 패키지 클래스로부터 접근 가능

다이어그램 도구



2 Class Diagram

4 Stereo Type

1 역할

UML 제공 기본요소 이외의 추가적인 확장 요소

“<< name >>” 으로 표기

2 종류

<<abstract>> Phone
+ owner:String - number:String
+Phone()<<create>> +PowerOn():void{abstract} +PowerOff():void{abstract}

<<interface>>

인터페이스 클래스

<<abstract>>

추상화 클래스

<<enumeration>>

열거형 타입 클래스

<<utility>>

인스턴스가 없는 static 메소드만 모아둔 클래스

<< create>>

생성자

다이어그램 도구



2 Class Diagram

5 Relationship



Relationship (관계)

모델 요소 간의 논리적 또는 물리적인 연결을 의미

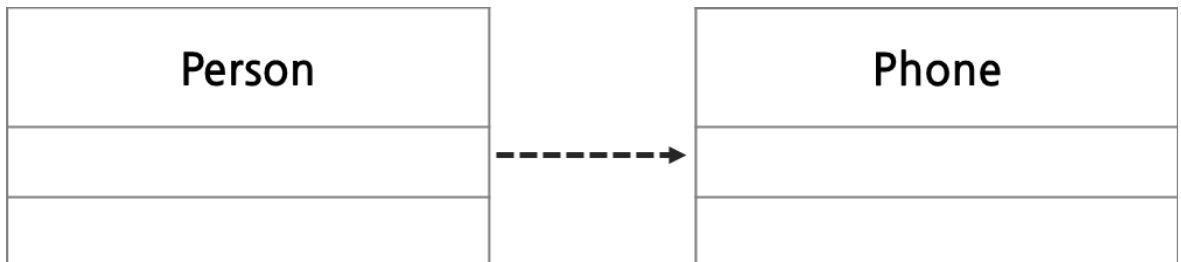


여러 객체의 유기적인 관계를 통해 시스템이 실행됨

1 의존관계(Dependency)

하나의 모델 요소가 다른 모델 요소를 사용하는 관계

‘---->’로 표시되며
화살표의 방향은 사용하는 쪽에서 사용되는 쪽으로 함



다이어그램 도구



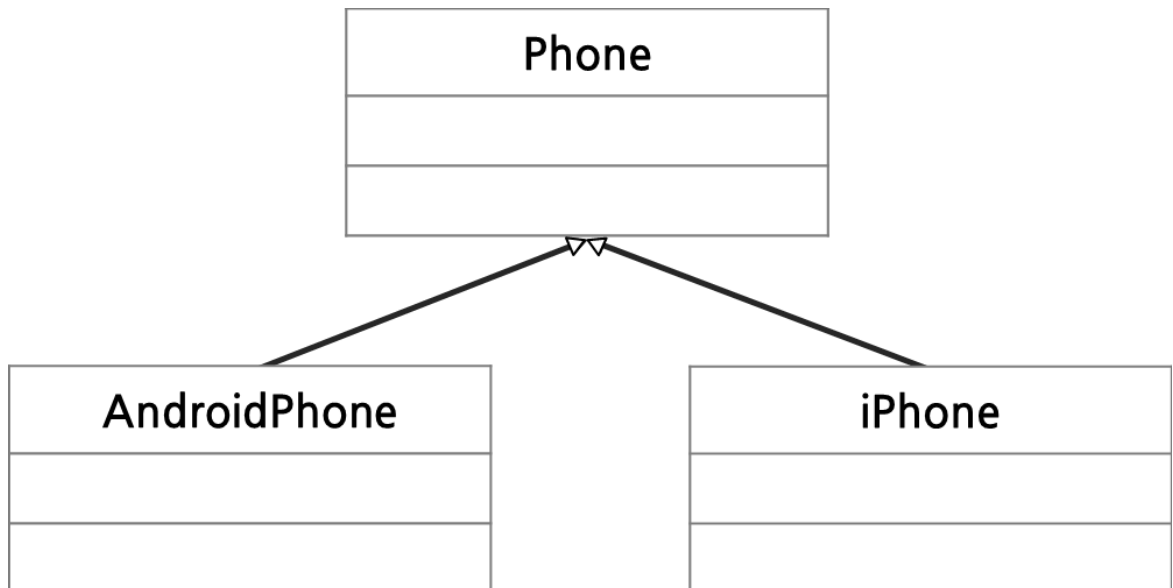
2 Class Diagram

5 Relationship

2 일반화(Generalization)

여러 클래스가 가진 공통적인 특징을 추출하여
공통적인 클래스를 일반화시키는 것

클래스 간 'is-a' 관계이어야 하며, 객체 지향의 상속 관계를 의미



다이어그램 도구



2 Class Diagram

5 Relationship

3 연관관계(Association)

관계 클래스로부터 생성된 **인스턴스들 간의 관계**를 표현

연관관계의 방향의 종류는 **양방향**과 **단방향**이 있음

Phoneltem	Role name ex) 핸드폰 상품	Role name ex) 안드로이드폰	AndroidPhone
	인스턴스수	인스턴스수	
	ex) 1 : 1개		
	0..1 : 0 혹은 1개		
	* : 다수		
	1..* : 1개 혹은 다수		

다이어그램 도구



2 Class Diagram

5 Relationship

4 집합연관관계(Aggregation)

전체와 부분을 나타내는 모델요소

전체를 나타내는 클래스와 이를 이루고 있는 부분 클래스의 관계를 나타냄

'has a' 관계를 나타냄, 전체와 부분은 서로 독립적인 관계



다이어그램 도구



2 Class Diagram

5 Relationship

5 복합연관관계(Composition)

전체와 부분을 나타내는 모델요소

전체를 나타내는 클래스와 이를 이루고 있는 부분 클래스의 관계를 나타냄

'has a' 관계를 나타냄, 클래스는 생명주기가 같음




프로세스 모델링 도구

1 프로세스 모델링


1 개요

1 프로세스 모델링이란?



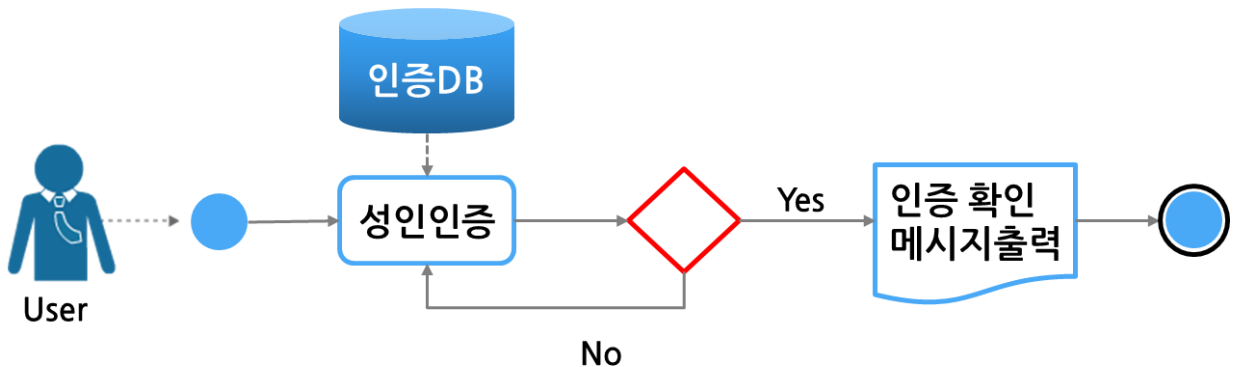
프로세스 모델링

모든 엔티티 집합 및 그것들의 관계나 속성들과 무결성 규칙들은 관련된 업무의 규칙을 발견하기 위하여 프로세스 모델을 사용



업무 프로세스가 쓰는 데이터가 무엇이며 그것들의 규칙이 어떤 것인지 알기 위하여 작성

2 프로세스 모델링 예 : 성인 인증 프로세스



프로세스 모델링 도구



1 프로세스 모델링

2 프로세스 모델링 법칙

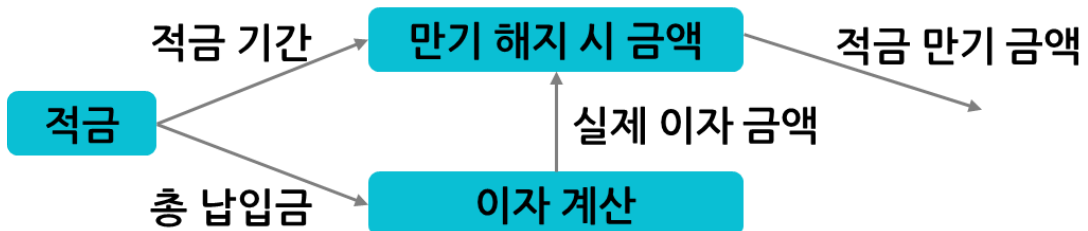
1 데이터 보존의 법칙(Conservation Rule)

입력된 데이터로부터 어떤 처리 과정을 거쳐도 그 데이터의 본질은 그대로 남음



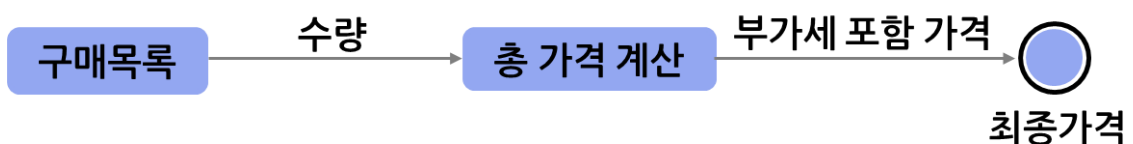
2 최소 절약의 법칙(Parsimony)

하나의 프로세스가 출력을 내는 데 있어서 필요한 최소의 데이터만을 입력받아야 함



3 지속성(Persistence) 법칙

입력과 출력의 정의가 바뀌지 않는 이상은 입력이 들어오면 항상 수행할 준비가 되어 있어야 함



프로세스 모델링 도구



1 프로세스 모델링

2 프로세스 모델링 법칙

4 순차처리(Order)의 법칙

데이터 저장소로부터 입력되는 데이터에 대한 처리는 입력되는 순서에 따라서 처리

데이터 저장소로의 접근은 어떠한 순서에 의해서 접근해도 괜찮음

예 : 은행 입출금 프로세스

5 독립성(Independence)과 영구성(Permanence)의 법칙

독립성(Independence)

- 다른 프로세스에 영향을 받지 않음

영구성(Permanence)

- 데이터 저장소에 저장된 데이터는 없어지지 않음


프로세스 모델링 도구



2 BPMN(Business Process Modeling Notation)

1 BPMN(Business Process Modeling Notation)

1 개념



BPMN

- 업무 프로세스 플로우와 웹 서비스를 모델화한 새로운 기준
- KAISHA-Tec을 위한 핵심 표준으로 업무 프로세스 다이어그램(Business Process Diagram-BPD)을 서식화하는 **완벽한 프로세스 모델링 기호법**

2 목적

1 비즈니스 분석가로부터 모든 비즈니스 사용자까지 즉시 이해할 수 있는 표기법 규정

2 초기 프로세스 초안 생성, 기술 개발자와 비즈니스 사용자를 위해 만들어짐

3 CASE tool

Visio 연계툴(스텐실)	개발툴
<ul style="list-style-type: none"> PRESY & BPMN Visio Plugin(Opensource) 	<ul style="list-style-type: none"> Together Architect 2006 for <u>Eclipse</u> (www.borland.com) Jviews Diagrammer (www.ilog.co.jp) Oracle BPEL (www.oracle.com)

이클립스와 연계 개발

프로세스 모델링 도구



2 BPMN(Business Process Modeling Notation)

2 BPMN 기본요소

1 플로오브젝트

이벤트

시작(개시) 이벤트, 각 프로세스의 시작을 나타냄

중간 이벤트

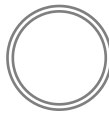
개시 이벤트로부터 종료 이벤트 사이에 발생

종료 이벤트

각 프로세스가 종료하는 곳을 나타냄



시작이벤트



중간이벤트



종료이벤트

2 트리거 : 이벤트가 시작하는 방법, 요인을 표기



없음
(None)

트리거가 없음



메시지
(Message)

도달 메시지가 프로세스의 시작하는 트리거가 됨



타이머
(Timer)

특정한 일시 또는 주기가 프로세스의 시작 트리거가 됨

프로세스 모델링 도구



2 BPMN(Business Process Modeling Notation)

2 BPMN 기본요소

2 트리거 : 이벤트가 시작하는 방법, 요인을 표기



룰
(Rule)

조건, 지정한 조건이 참일 때 시작 트리거가 됨



링크
(Link)

프로세스가 종료되려 할 때 다른 새로운 프로세스의 시작으로 전이시킬 때



복합
(Multiple)

프로세스의 트리거가 될 방법이 다수 존재할 때



에러
(Error)

예외가 발생할 때 처리를 하기 위한 트리거



취소
(Cancel)

트랜잭션과 관련되어 사용



보상
(Compensation)

취소와 비슷하나 트랜잭션과 묶이지 않는 개별 스텝

프로세스 모델링 도구

2 BPMN(Business Process Modeling Notation)

2 BPMN 기본요소

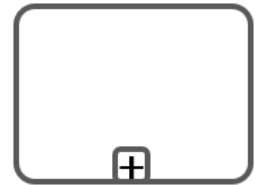
3 액티비티, 마카

태스크, 서브프로세스

- 서브 프로세스는 거대한 비즈니스 프로세스를 일반화시켜 개략적으로 나타냄



Task



Sub process

마카

- 태스크, 서브 프로세스를 Loop, 비동기, 보상 등의 관점에서 상세히 기술

Task

Sub-process

Loop



Multiple Instance



Ad-hoc



Compensation



학습정리

1. 다이어그램 도구



- 모델링 관련 도구로는 UML(Unified Modeling Language), UP(Unified Process)가 있음
- 통합 개발 환경(IDE) 도구는 Visual Studio, Eclipse와 같은 개발 툴이 있음
- DBMS 도구로는 mysql, oracle, mssql, postgresql, MongoDB 등이 있음
- Class Diagram은 클래스와 그들 간의 관계를 정의함
- 정적인 관점에서 클래스 구조를 표현하며 데이터 모델링을 수행하는 용도도 포함됨
- 코딩을 위한 직접적인 설계 사양을 제공하며 논리적·물리적인 관점까지 일관된 형식으로 시스템을 분석, 설계하는 도구로 사용됨

학습정리

2. 프로세스 모델링 도구



- 프로세스 모델링은 모든 엔티티 집합 및 그것들의 관계나 속성들과 무결성 규칙들은 관련된 업무의 규칙을 발견하기 위하여 프로세스 모델을 사용함
- 프로세스 모델링을 위한 법칙으로 총 6가지의 법칙이 있음
 - 1) 데이터 보존의 법칙
 - 2) 최소 절약의 법칙
 - 3) 지속성 법칙
 - 4) 순차 처리의 법칙
 - 5) 독립성의 법칙
 - 6) 영구성의 법칙
- BPMN(Business Process Modeling Notation)은 업무 프로세스 플로우와 웹 서비스를 모델화한 새로운 기준이며, 비즈니스 분석가로부터 모든 비즈니스 사용자까지 즉시 이해할 수 있는 표기법을 규정하는 목적을 가짐