학습 목표

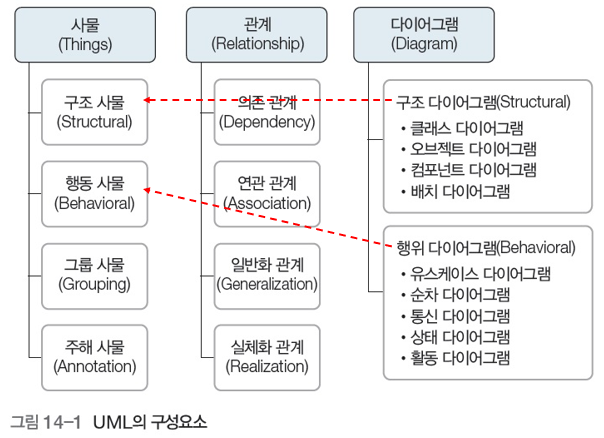
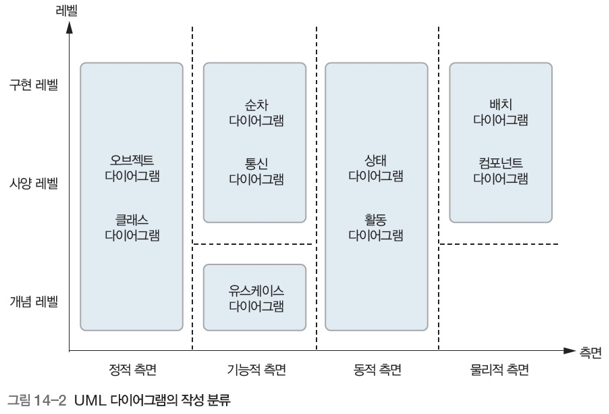
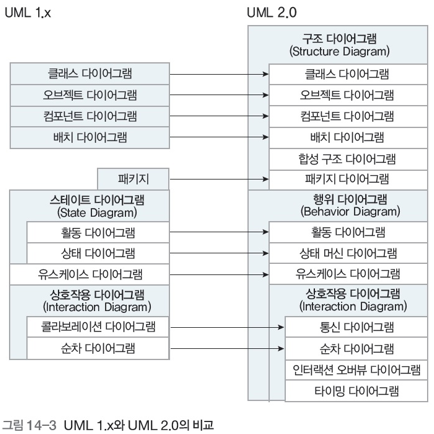
UML의 구성요소를 이해한다.

UML의 정적 모델링 도구를 학습한다

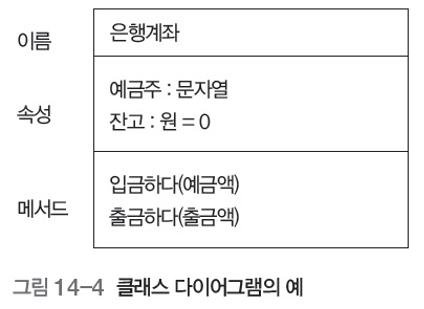
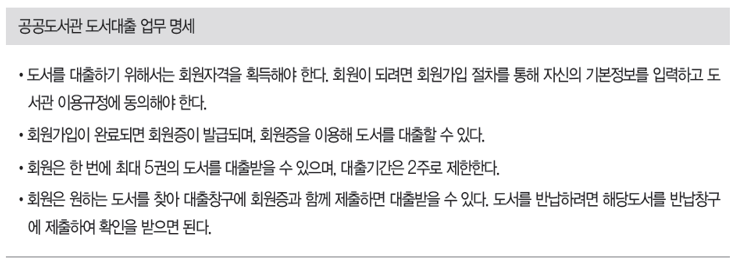
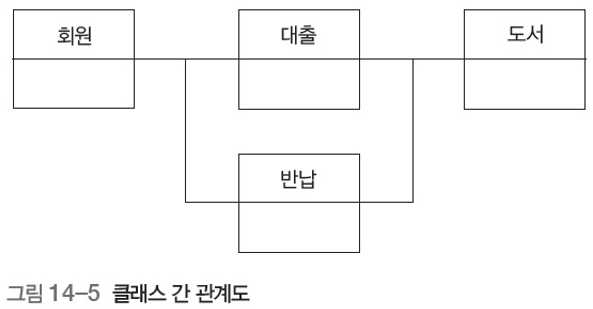
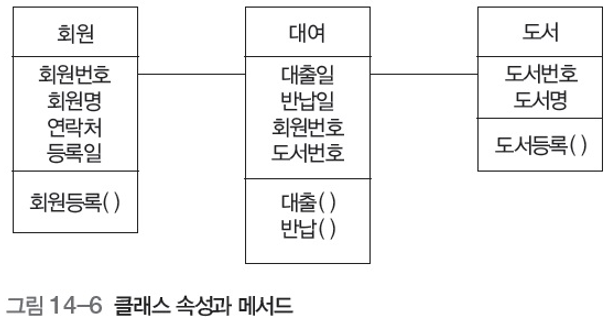
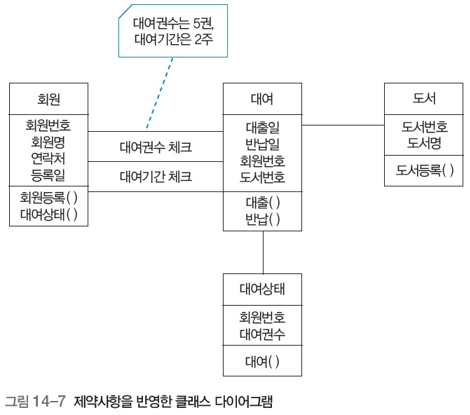
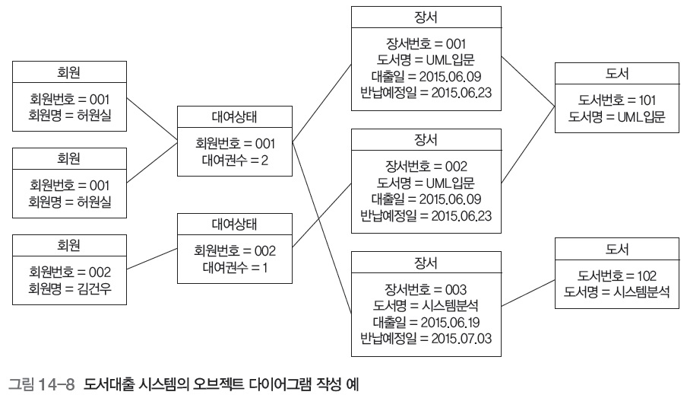
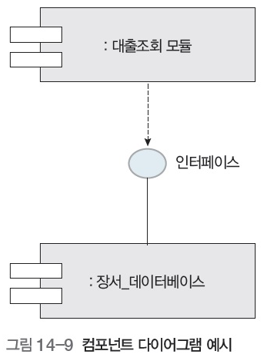
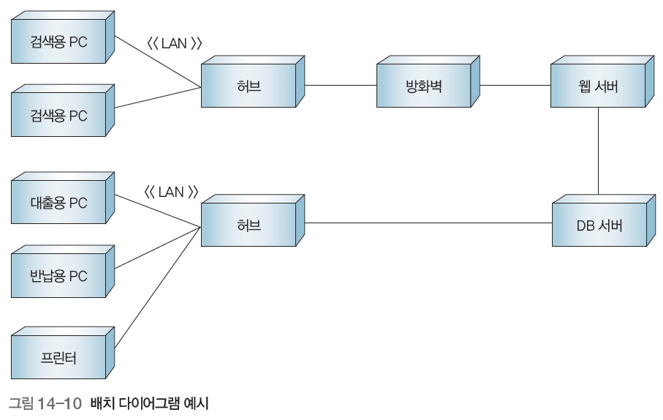
UML의 동적 모델링 도구를 학습한다.

UML을 활용한 분석 설계 과정을 이해한다.

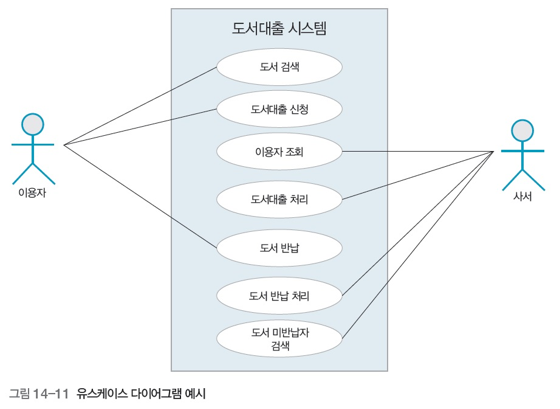
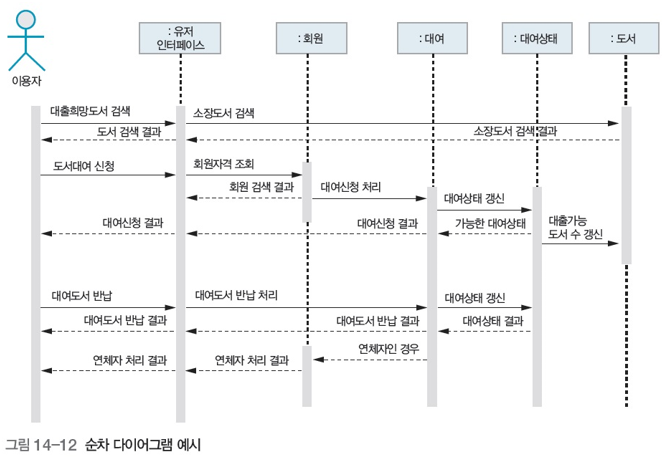
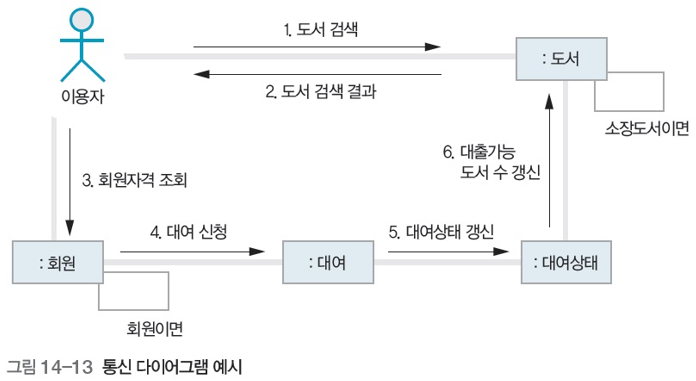
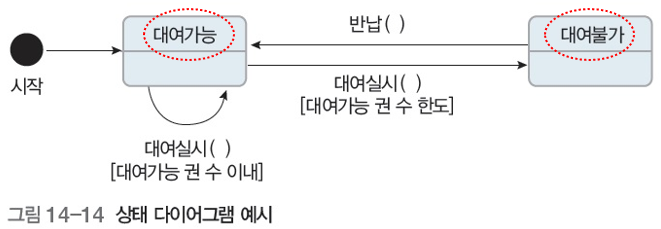
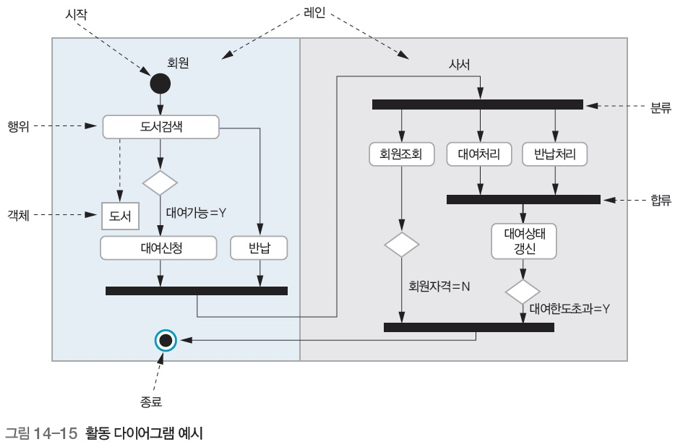
**14.1 UML의 구조**

1. UML(Unified Modeling Language)의 특징
   * 가시화: 심벌에 명확한 정의를 부여하여 시각적인 그래프로 표기
   * 명세화: 정확, 명백, 완전한 모델 작성
   * 구축: 다양한 프로그래밍 언어로 generation이 가능
   * 문서화: 업무분석에 대한 Documentation이 가능하고  
     요구사항, 상세내역, 테스트 단계의 언어도 제공
2. UML의 구성요소  
   
   * 사물
     1. 구조 사물(Structural Things): 시스템의 구조  
        Class, Use Case, Interface, Communication, Active Class, Component, Node
     2. 행동 사물(Behavioral Things): 시스템의 행위  
        교류(Interaction), 상태 머신(State Machine)
     3. 그룹 사물(Grouping Things): Package
     4. 주해(주석) 사물(Annotation Things): Node
   * 관계
     1. 의존 관계(Dependency Relationship)  
        예) TV와 리모컨
     2. 연관 관계(Association Relationship): ‘has-a’ 관계  
        예) 자동차와 부품
     3. 일반화 관계(Generalization Relationship): ‘is-a’ 관계  
        예) 부모 클래스로서의 자동차와 자식 클래스로서의 택시, 버스, 트럭 등
     4. 실체화 관계(Realization Relationship)  
        한 객체가 다른 객체에 의해 오퍼레이션을 수행하도록 지정하는 것  
        예) TV의 행동 중 일부가 리모컨의 행동을 ‘실체화(Realize)’ 함  
        클래스(TV)와 인터페이스(리모컨)가 가지는 관계가 실체화 관계
   * 다이어그램
     1. 구조 다이어그램: 정적 모델링을 위한 다이어그램  
        클래스, 오브젝트, 컴포넌트, 배치
     2. 행위 다이어그램: 동적 모델링을 위한 다이어그램  
        유스케이스, 순차, 통신, 상태, 활동
3. UML 모델링의 이해
   * 네 가지 측면
     1. 정적 측면: 모델링 대상 범위에서 객체의 구조와 관계를 나타냄
     2. 동적 측면: 이벤트가 발생함에 따라 객체의 상태 변화를 나타냄
     3. 기능적 측면: 사용자의 측면에서 본 시스템의 행동을 나타냄
     4. 물리적 측면: 시스템 실행을 위해 필요한 컴퓨터와 기억 매체의 공간적인 배치 기술
   * 세 가지 레벨
     1. 개념 레벨: 문제 영역(도메인)의 해석을 기록
     2. 사양 레벨: 설계 작업
     3. 구현 레벨: 개발 작업
   * UML 다이어그램의 작성 분류  
     
   * UML 1.x와 UML 2.0의 비교  
     
4. 정적 모델링 도구
   * UML의 구조 다이어그램에 해당
   * 시스템의 지속적이고 정적인 측면을 모델링하는 특징이 있음

**14.2 정적 모델링 도구**

1. 클래스의 구성요소
   * 이름
   * 속성
   * 메서드(오퍼레이션)  
     
2. 클래스 다이어그램 작성 절차
   1. 클래스 명세를 결정하고.
   2. 메서드(오퍼레이션)를 찾아낸 후,
   3. 필요한 속성을 추출한다.
3. 클래스 다이어그램 작성 예시  
   
   1. 클래스 후보 추출: 클래스 후보를 추출하기 위한 가장 일반적인 방법은 업무 명세서에서 명사를 찾아내는 것  
      예) 회원, 회원가입, 회원증, 도서, 대출, 반납
   2. 클래스와 클래스 사이의 관계 규정  
      
   3. 클래스의 메서드와 속성 규정  
      
   4. 클래스의 메서드와 속성 규정 (개선 후)  
      
4. 오브젝트 다이어그램
   * 특정 시점의 오브젝트들의 구조적 상태를 표현  
     
5. 컴포넌트 다이어그램
   * 인터페이스에 의해서 기능이 정의된, 독립적으로 개발・배포・조립 가능한 시스템의 구성단위 예) J2EE 플랫폼의 JAR 파일, 닷넷 플랫폼의 DLL 파일  
     
6. 배치 다이어그램
   * 노드를 입체적으로 표현하고, 그 사이를 의존 화살표와 접속 관계를 나타내는 실선으로 연결해 이들 간의 통신 관계를 나타낸 것.
   * 노드: 시스템 실행 시 필요한 물리적인 처리 자원과 실행 모듈, 소프트웨어 컴포넌트의 인스턴스 등을 뜻함.
7. 배치 다이어그램의 작성 예시  
   

**14.3 동적 모델링 도구**

1. 동적 모델링 도구란?
   * UML의 행위 다이어그램과 인터랙션 다이어그램에 해당
   * 시간의 흐름에 따라 유동적으로 변하는 객체의 상태나 행위, 객체 간의 상호작용 등을 표현
2. 유스케이스 다이어그램
   * 액터(Actor)의 관점에서 본 시스템의 기본적인 행동을 기술한 것  
     
3. 순차 다이어그램
   * 정의: 객체 간의 동적 상호작용을 시간의 흐름에 따라 나타낸 것
   * 특징
     1. 객체의 메서드(오퍼레이션)와 속성을 상세히 정의
     2. 객체의 책임: 순차 다이어그램의 객체는 다른 객체가 의뢰하는 일을 처리
     3. 유스케이스를 실현
4. 순차 다이어그램의 예시  
   
5. 통신 다이어그램
   * 객체들 사이에 주고받는 메시지를 표현  
     
6. 상태 다이어그램
   * 객체의 상태가 이벤트의 발생 혹은 시간의 경과에 의해 어떻게 변화하는지를 나타낸 것
   * 특정 객체가 생성하여 소멸할 때까지의 라이프 사이클을 모델화함  
     
7. 활동 다이어그램
   * 예전부터 사용해왔던 순서도(Flow Chart)와 모양이 매우 비슷
   * 순서도(어떤 행위에 대한 워크플로우를 표현하는 대표적 수단)를 객체지향 스타일로 개선시킨 형태
8. 활동 다이어그램의 예시  
   

**14.4 UML 모델링 절차**

1. UML 모델링의 절차
   1. 초기 클래스 다이어그램 작성 < 업무명세서
   2. 유스케이스 다이어그램 작성 < 사용자 요구사항
   3. 클래스 다이어그램 변경 < 유스케이스 다이어그램
   4. 순차 다이어그램과 통신 다이어그램 작성 < 유스케이스에 해당하는 기능
   5. 인터페이스를 식별(클래스 다이어그램 변경) < 메시지 교환, 시간적 순서
   6. 활동 다이어그램과 상태 다이어그램을 작성 < 동적 모델링 고려
   7. 컴포넌트 다이어그램 및 배치 다이어그램 작성 < 최종구현과 설치 고려