객체지향소프트웨어공학

9주차 : UML 기반 시스템 분석 설계

교재#2 : I-06. 분석 모델링

1. 분석 모델의 의의
2. 분석 모델의 구성 요소
3. 분석 클래스 모델
4. 분석 모델 검토
5. 한국 IT 대학 분석 모델
6. 관련 산출물 사례 및 작성 지침
7. 요약

6장 학습목표

1. 분석 모델에 대한 설명
2. 분석 모델의 구성요소 이해
3. 분석 모델 작성 방법 설명
4. 관련 산출물 작성 방법 설명 및 이해
5. 교재 P52~56의 분석설계 워크플로우에 대한 상세 및 사례 설명

분석 모델의 의의

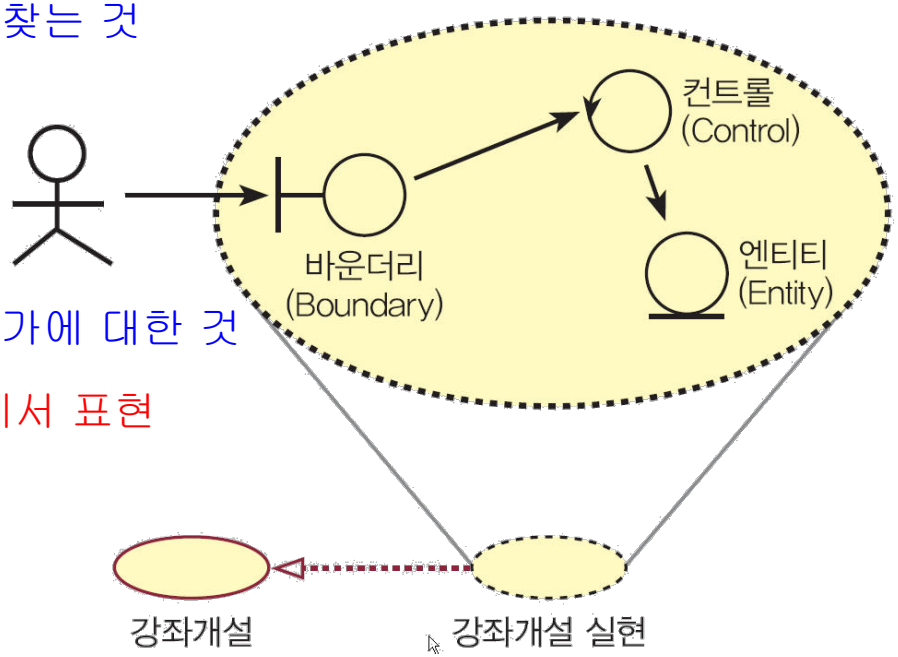
- 유스케이스 실현(Realization)
  - 유스케이스의 실현 시나리오에 참여하는 것으로 클래스 다이어그램과 시퀀스 다이어그램으로 표현
- 시나리오
  - 유스케이스를 실현하기 위해서 필요한 객체, 클래스, 객체들 간의 관계를 발견하는 것을 돕기 위해 사용
  - 어떻게 유스케이스에서 기술된 책임이 시스템의 객체와 클래스에 분산되는가에 대한 것
  - 사용자, 도메인 전문가의 입장에서 기술
  - 고객이나 도메인 전문가들이 개발자에게 그들이 원하는 시스템의 행위에 대해서 말할 수 있는 수단 제공

분석 모델의 의의

- 시나리오 작성 및 정제
  - 1차 (Primary) 시나리오
    - 무엇을 해야 하는지 기술
  - 2차(Secondary) 시나리오
    - 조건에 의해 분기되는 부분에 대한 기술
- UP에서 유스케이스 실현
  - 객체의 상호작용 표현으로 논리 뷰에서 작성
  - <<use-case realization>> 스테레오타입 적용
  - 실현되는 유스케이스의 이름과 같게 작성
  - 유스케이스 뷰의 유스케이스와 논리 뷰
    - 유스케이스와 유스케이스 실현 관계를 보여주는 다이어그램 작성
  - 실현 관계는 <<realize>> 스테레오타입을 갖는 의존 관계 사용

분석 모델의 구성 요소

- 분석 클래스
  - UP의 시스템을 보는 관점
    - 시스템 외부의 관점
      - 시스템을 사용하는 액터의 관점에서 시스템이 만족시켜 주어야 하는 요구사항들이 무엇이 있는지 찾는 것
      - 유스케이스 모델로 표현
    - 시스템 내부의 관점
      - 요구사항을 위해서 시스템이 어떠한 일들을 수행해야 하는가에 대한 것
      - 분석 모델을 거쳐 설계 모델에서 표현



분석 모델의 구성 요소

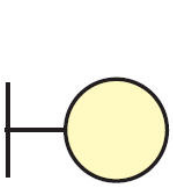
- 분석 클래스
  - 분석 모델
    - “무엇을 해야 하는가?”에서 “어떻게 해야 하는가?”로 넘어갈 때에 의미적 차이를 최소화하기 위해서는 중간 매개체 역할을 하는 것
  - 분석 모델의 작성
    - 유스케이스에 대한 텍스트 설명에 해당하는 사건 흐름을 사용자 관점에서 개발자 관점으로 정제하는 작업
    - 정제된 유스케이스의 사건 흐름으로부터 설계 모델에 사용될 초기 클래스 집합인 분석 클래스로 구조화

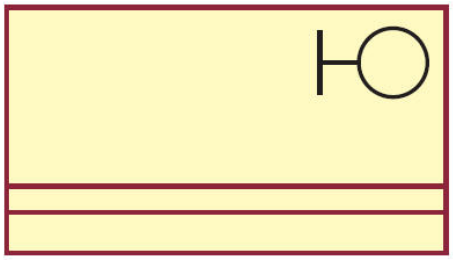
분석 모델의 구성 요소

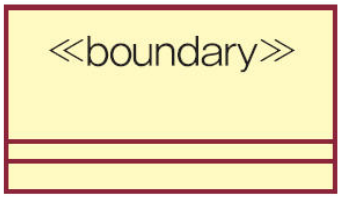
- 분석 클래스
  - 분석 클래스 구성
    - 경계 클래스(Boundary class)
      - 액터와의 상호작용 지원 역할
      - 외부에 존재하는 것들 (액터)에 의존적인 시스템의 부분 모델화
    - 컨트롤 클래스(Control class)
      - 경계 클래스와 엔티티 클래스 사이에 커뮤니케이션 역할
      - 유스케이스의 흐름 제어
      - 시스템의 외부에 존재하는 것에 독립적인 부분을 모델화
    - 엔티티 클래스(Entity class)
      - 도메인 객체 의미
      - 시스템의 외부에 존재하는 것에 독립적인 부분 모델화

분석 모델의 구성 요소

- 경계 클래스
  - 액터와의 상호작용을 지원하는 역할을 수행하는 분석 클래스
  - 경계 클래스의 다양한 표기법







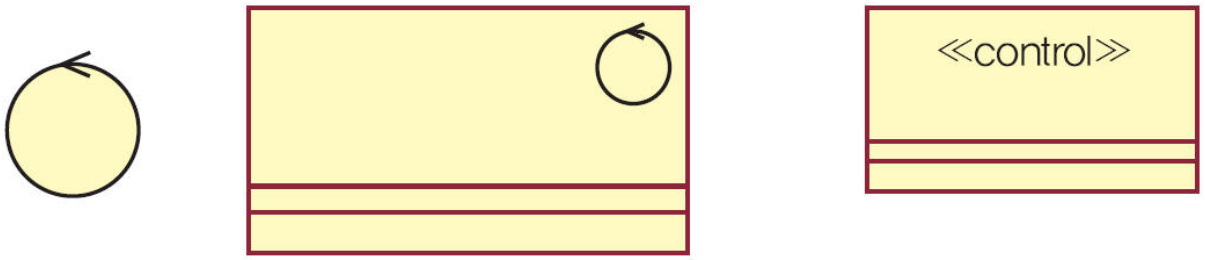
  - 시스템 인터페이스를 모델로 만드는 데 사용

분석 모델의 구성 요소

- 경계 클래스
  - 경계 클래스 분류
    - 액터가 사람일 경우
      - GUI(Graphic User Interface)와 같은 사용자 인터페이스
    - 시스템일 경우
      - 시스템 인터페이스
    - 사람 액터와 상호작용하는 경우
      - 유스케이스 모델에서 사용한 스토리보드
  - 경계 클래스 유형
    - 사용자 인터페이스 클래스
      - 시스템의 인간 사용자들의 커뮤니케이션을 돕는 클래스
    - 시스템 인터페이스 클래스
      - 다른 시스템과의 커뮤니케이션을 돕는 클래스
    - 장치 인터페이스 클래스
      - 센서와 같은 외부 사건을 감지할 수 있는 장치에 대한 인터페이스를 제공하는 클래스

분석 모델의 구성 요소

- 컨트롤 클래스
  - 하나 이상의 유스케이스에 특화된 일련의 행위 표현
  - 유스케이스에 기술된 행위를 실현하는데 요구되는 사건 흐름 제어
  - 순차적인 행위에 대해 하나 또는 그 이상의 경계 클래스와 엔티티 클래스들을 연결해 주는 역할 담당
  - 유스케이스를 '수행' 또는 '실행'하는 것으로 인식하게 만들고 유스케이스의 동적인 특성, 기능, 서비스를 구체적으로 표현
  - 애플리케이션에 의존적인 클래스
  - 컨트롤 클래스의 다양한 표기법

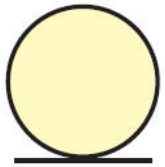


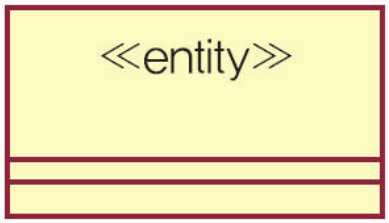
분석 모델의 구성 요소

- 컨트롤 클래스 (계속)
  - 유스케이스의 사건 흐름에 대해 책임을 갖기 때문에 정련 단계(Elaboration Phase)에서 아키텍처 구축 작업 초기에 각각의 유스케이스마다 컨트롤 클래스를 하나씩 만듦
  - 컨트롤 클래스의 사용은 매우 주관적
  - 유스케이스마다 컨트롤 클래스를 추가하는 것은 초기 산출물이고, 분석과 설계가 계속됨에 따라 컨트롤 클래스는 제거, 분리 또는 병합
  - UP에서의 컨트롤 클래스
    - 유스케이스의 흐름을 제어하는 역할
    - 특정 엔티티 클래스와 관련될 수 없는 업무로 논리 처리
    - 유스케이스 흐름 처리 부분과 업무 논리 처리 부분으로 분리

분석 모델의 구성 요소

- 엔티티 클래스
  - 실제 시스템이 처리해야 하는 일 담당
  - 일반적으로 오랫동안 존재되는 정보와 연관된 행위 표현
  - 주변 환경(액터)에 독립적
  - 업무 중심으로 추출
  - 엔티티 클래스의 다양한 표기법





분석 모델의 구성 요소

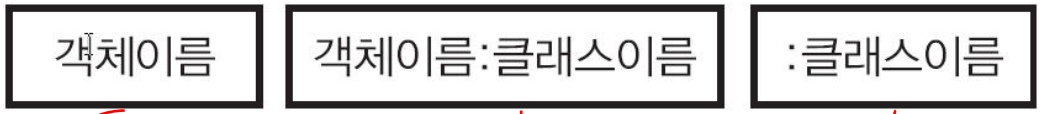
- 엔티티 클래스
  - 엔티티 클래스 찾기
    - 시스템이 수행해야 하는 일이 무엇인지 조사
    - 일반적으로 아키텍처구축 단계의 초기에 발견
    - 비즈니스 모델링 수행에 의한 비즈니스 객체 모델의 비즈니스 엔티티를 통해서 엔티티 클래스 찾음
  - 엔티티 객체(엔티티 클래스의 인스턴스)
    - 몇몇 현상(사건, 사람, 또는 몇몇 실세계의 객체)에 대한 정보를 갱신하고 유지하기 위해 사용
    - 일반적으로 오랜 기간 지속되어야 하며 속성과 관계 필요
    - 개발될 시스템의 핵심 개념 표현
  - 엔티티의 속성과 관계
    - 액터의 입력에 의해 주어지며, 시스템 내부작업 수행하기 위해 요구
    - 하나의 엔티티 클래스에 의해 독점된다면 속성이나 관계로 설계해야 하고 여러 엔티티 클래스에 의해 공유된다면 엔티티 클래스로 설계

분석 모델의 구성 요소

- 인터행위 다이어그램
  - 분석 모델에서 인터행위 다이어그램(Interaction diagram)의 작성 목적은 유스케이스의 책임을 객체들의 책임으로 분산시키는 것
- 인터행위 다이어그램 작성 목적
  - 클래스의 연산을 찾는 것
  - 구현 클래스들 간의 상호작용을 시간적인 순서로 작성함으로써 구현 시 코드 작성에 사용
  - **컴뮤니케이션 다이어그램, 시퀀스 다이어그램 사용 권장**

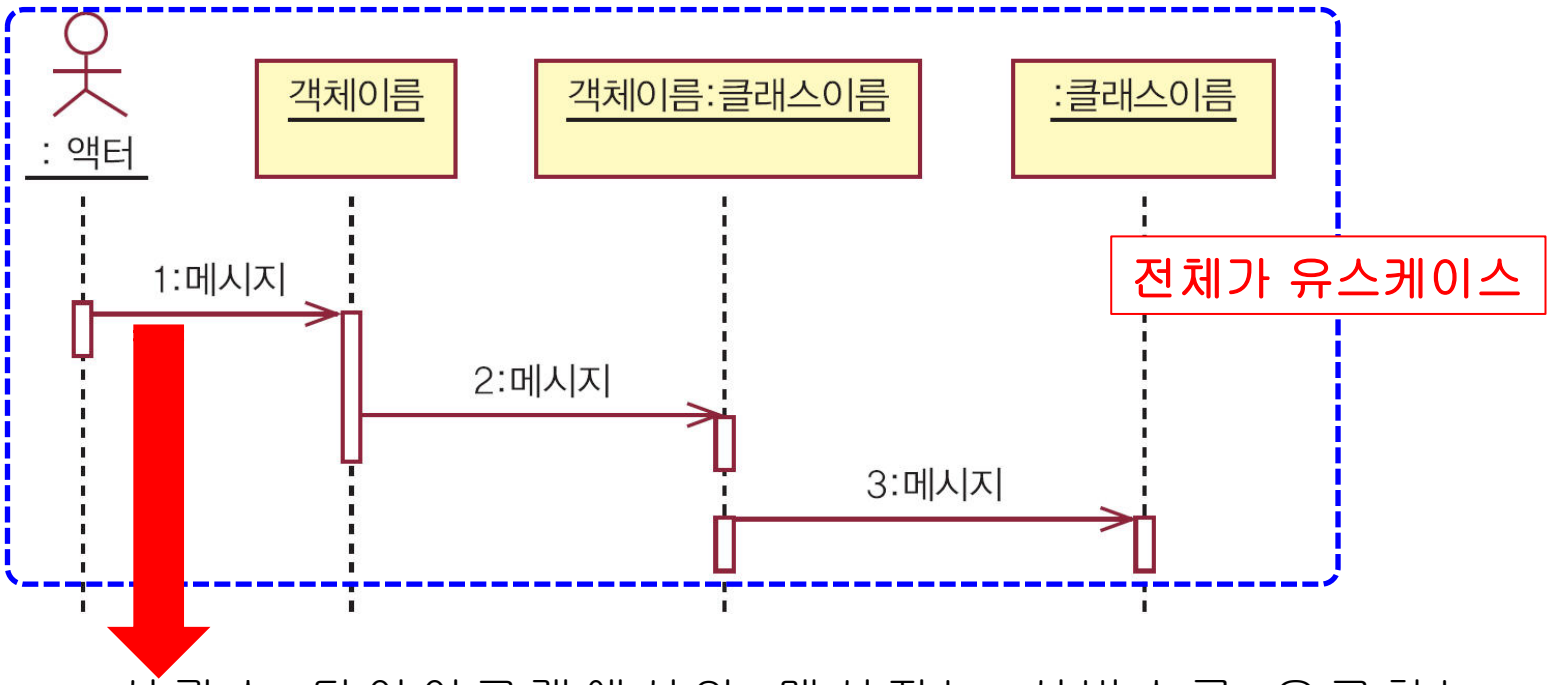
분석 모델의 구성 요소

- 시퀀스 다이어그램(Sequence Diagram)
  - 시간적인 순서로 정렬된 객체들의 상호작용 표현
  - 시나리오에 포함된 객체, 클래스와 시나리오의 기능을 수행하기 위해 필요한 객체, 클래스와 그들 사이에 교환되는 메시지의 시퀀스 설명
  - 시퀀스 다이어그램에 작성되는 객체들은 밑줄이 쳐진 객체의 이름을 포함하는 사각형으로 작성
  - 시퀀스에서의 객체 이름 작성



    - 1번째 형식
      - 베이스 클래스는 규정하지 않고 객체 이름만 지정
    - 2번째 형식
      - 베이스 클래스로부터 생성된 특정 객체 지정
    - 3번째 형식
      - 베이스 클래스만 규정함으로써 그 클래스로부터 생성된 모든 객체 지정

- 시퀀스 다이어그램(Sequence Diagram)
  - 시퀀스 다이어그램에서의 객체와 메시지에 대한 UML 기호

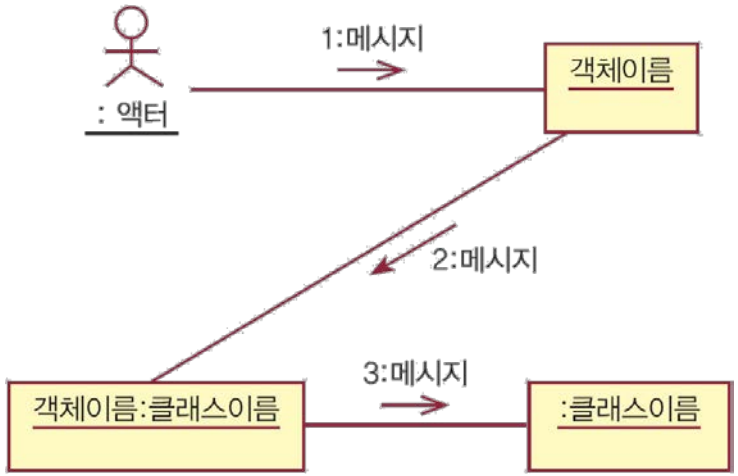


  - 시퀀스 다이어그램에서의 메시지는 서비스를 요구하는 객체(메시지를 보내는 객체)로부터 서비스를 제공하는 객체(메시지를 받는 객체)로 화살표로 표현

분석 모델의 구성 요소

- 컴뮤니케이션 다이어그램(Communication Diagram)
  - 시나리오를 보여주기 위한 선택적인 방법
  - 객체와 다른 객체와의 연결
  - 조직화된 객체들 간의 상호작용 표현
  - 컴뮤니케이션 다이어그램의 포함 사항
    - 사각형으로 그려진 객체들
    - 객체와 객체 사이의 링크
    - 텍스트와 화살표로써 표시되는 메시지
  - 컴뮤니케이션 다이어그램에서의 객체 링크 메시지에 대한 UML 기호

시이퀀스 다이어그램 ⇔ 컴뮤니케이션 다이어그램!



분석 클래스 모델

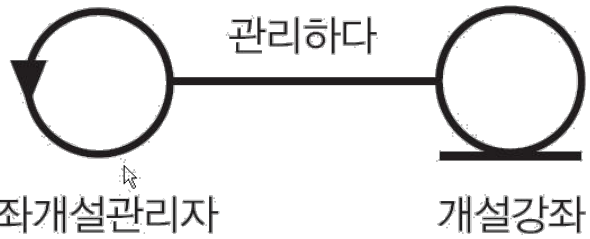
- 객체 모델
  - 특별한 시스템이나 애플리케이션을 모델화하기 위해 만들어지는 객체들을 설명하기 위해 사용되는 개념의 집합
- 관계 작성
  - 객체들의 메시지 전달
    - 객체들 사이의 관계를 통해서 전달
    - 객체 사이의 관계 찾기
      - 객체 사이에 메시지 전달이 존재하는 객체들을 조사
  - 클래스 관계 분석

송신 객체	수신 객체	관계 종류
교수정보UI	교수정보관리자	연관
교수정보관리자	교수	연관

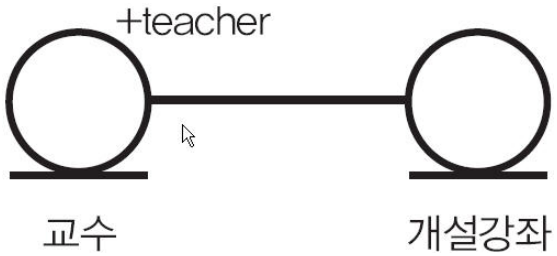
분석 클래스 모델

- 관계 작성
  - 객체들의 관계 파악
    - 객체들이 사용되는 문제 영역 중심
  - **집합 관계**
    - “part-of” 또는 “containment” 관계
  - 연관 & 집합 관계를 위한 UML 표기

분석 클래스 모델



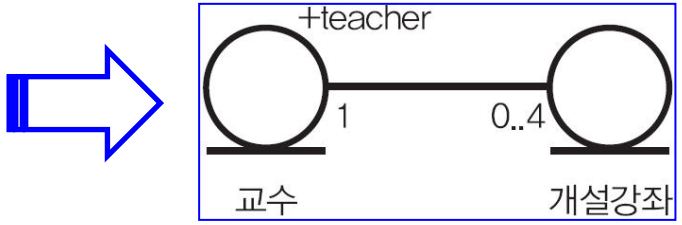
- 관계 작성
  - 관계 명명
    - 이름은 일반적으로 관계의 의미를 전달할 수 있는 능동의 동사구
    - 동사구는 일반적으로 읽는 방향을 암시하기 때문에 왼쪽에서 오른쪽으로 위에서 아래로 올바르게 읽게 하도록 연관의 이름 명명
  - 역할(Role) 명명
    - 클래스에 연결된 연관의 끝 부분에 작성
    - 연관 이름 대신에 사용 가능



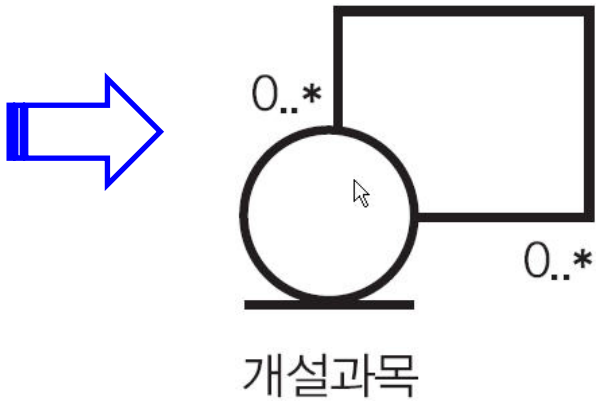
    - 역할의 이름은 연관된 두 클래스 사이에 존재하는 서로에 대한 자격을 의미하는 명사로 작성
    - 역할 이름을 찾을 때는 역할 이름을 작성할 상대 객체의 입장에서 해당 객체 바라봄

분석 클래스 모델

- 관계 작성
  - 다중성(Multiplicity)
    - 관계에 참여하는 객체의 수 정의
    - 상호 링크되어 있는 객체의 수 표현
    - 일반적인 다중성 표현
      - 1 : 정확히 1개 의미
      - 0..* : 없을 수도 있고 여러 개 있을 수 있음
      - 1..* : 하나 또는 그 이상을 의미
      - 0..1 : 하나도 없거나 하나가 존재함을 의미
      - 5..8 : 특별한 범위를 나타냄 (5, 6, 7, 8)
      - 4..7.9 : 조합을 나타냄(4,5,6,7이거나 또는 9)
    - "개설 강좌 객체에 대해 교수 객체는 'teacher'의 역할을 갖는다"
    - "하나의 개설 강좌 객체는 정확히 하나의 교수 객체와 관련된다"
    - "하나의 교수 객체는 0에서 4개까지의 개설 강좌의 객체들과 관련된다."

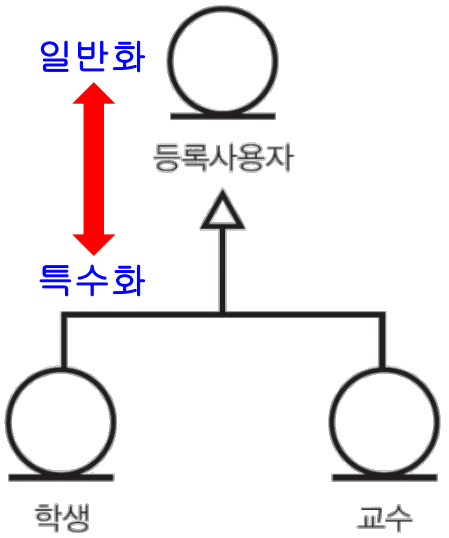


- 관계 작성
  - 재귀(Reflexive)관계
    - 같은 클래스에 속한 다수의 객체들이 서로 커뮤니케이션을 하는 경우
    - 일반적으로 연관 관계 이름보다는 역할 이름 사용
    - 선행과목의 역할을 수행하는 하나의 개설과목 객체는 0개 이상의 개설과목 객체들과 관련
    - 하나의 과목 객체는 선행과목으로 역할을 수행하는 0개 이상의 개설과목 객체들과 관련



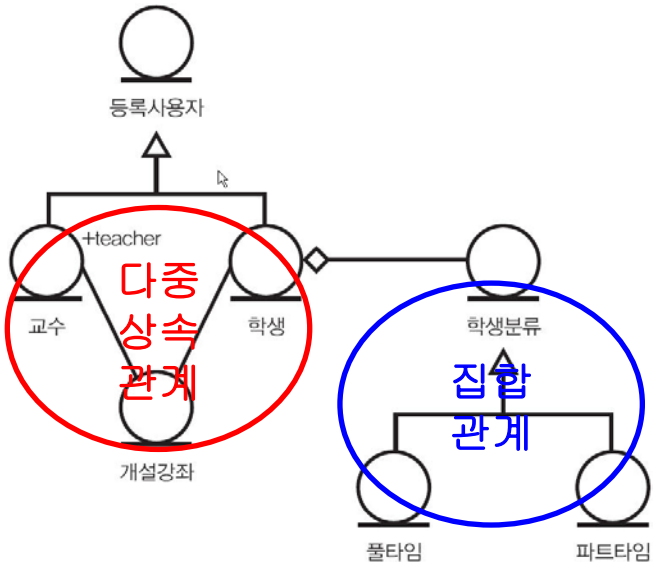
분석 클래스 모델

- 관계 작성
  - 상속(Inheritance)관계
    - 하나의 클래스가 하나 또는 다수의 클래스의 구조와 행위를 공유하는 경우에 발생하는 클래스 사이의 관계
    - "is-a" 또는 "kind-of" 관계로 불림
    - 관계 이름, 역할 이름, 다중성은 적용되지 않음
    - 일반화(Generalization)
      - 여러 클래스들에서 존재하는 공통적인 구조와 행위를 캡슐화하는 상위 클래스를 만들 수 있는 능력 제공
      - 일반화는 분석 초기에 행해지는 공통되는 노력 포함
      - 클래스들은 구조(속성)와 행위(연산)의 공통성에 대해서 조사
    - 특수화(Specialization)
      - 상위 클래스에 대한 구체화를 나타내는 하위 클래스를 만드는 능력 제공
      - 일반적으로 구조와 행위는 새로운 클래스에 추가
      - 상속의 발견 방법은 클래스가 이미 존재할 때 사용하는 것으로, 하위 클래스가 기존 클래스 행위를 특수화 할 필요가 있을 경우에 추가

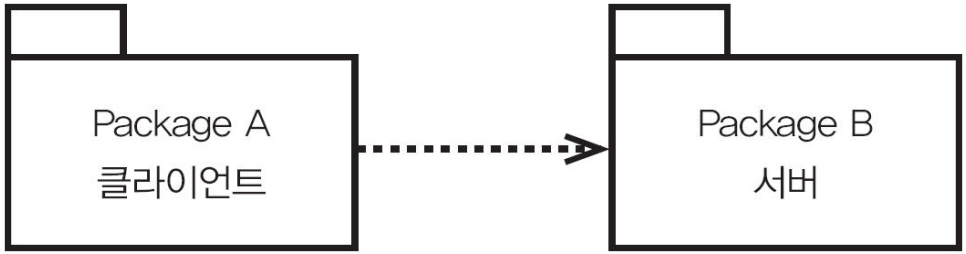


분석 클래스 모델

- 관계 작성
  - 상속(Inheritance)관계 (계속)
    - 단일 상속
      - 상위 클래스를 하나만 갖는 것
    - 다중 상속
      - 상위 클래스를 여러 개 갖는 것
      - 실제 프로그램으로 옮겼을 때는 많은 관련 문제들을 야기할 수 있음
  - 상속 대 집합 관계



- 관계 작성
  - 패키지(Package) 관계
    - 패키지 간의 관계 유형은 의존 관계



    - 의존 패키지를 향해 점선 화살표로써 표현
    - 패키지 A = 클라이언트 패키지
    - 패키지 B = 서버 패키지
    - 시스템의 시나리오나 클래스 관계에서 식별 가능
    - 반복적인 과정을 통해 관계들은 분석 및 설계가 진행됨에 따라 변화

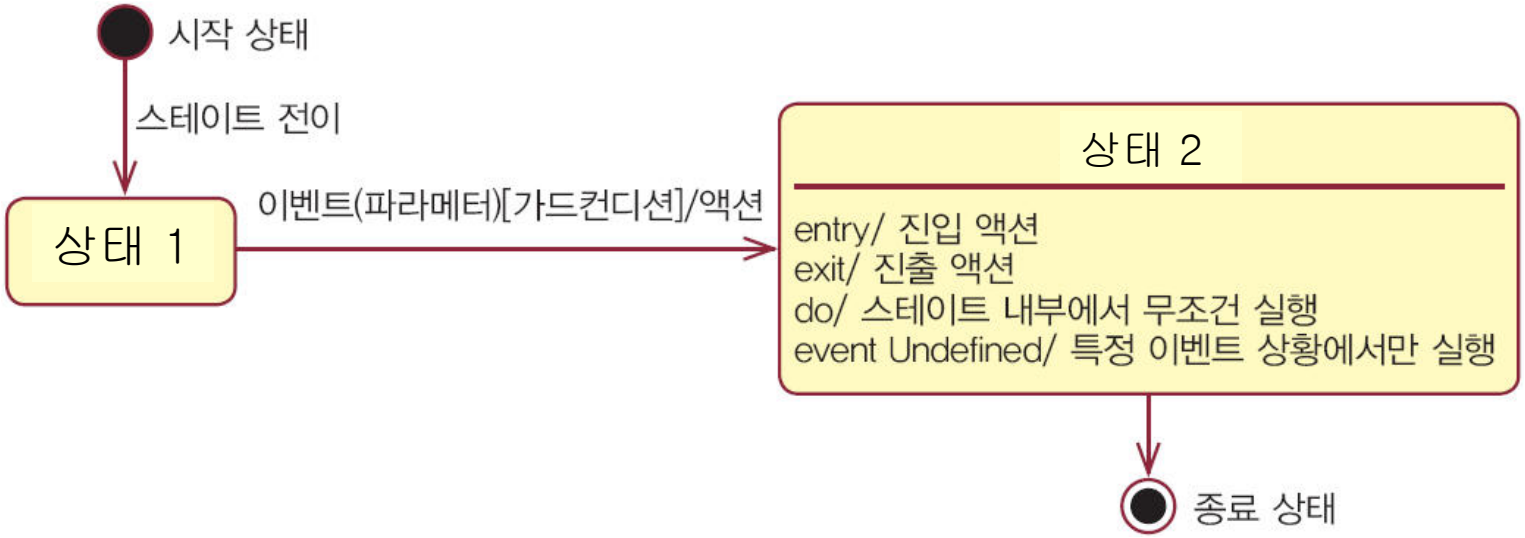
분석 클래스 모델

- 객체의 구조와 행위 작성
  - 좀 더 읽기 쉽고 유지하기 쉬운 연산을 정의할 때 스타일 사용
    - 스타일 사용은 클래스 전체에 걸쳐 일관성 제공
    - 모델과 코드를 더 쉽게 읽고 관리할 수 있게 함
  - 연산(Operation)
    - 인터행위 다이어그램인 시퀀스 다이어그램과 컴뮤니케이션 다이어그램을 이용해서 식별
    - 연산의 이름은 연산 수행을 요구하는 클래스의 관점이 아닌 연산을 수행하는 클래스 관점에서 명명
    - 분석 시(논리적) “~요청”으로 작성
    - 설계 시(물리적) 구현언어에 맞게 변경

분석 클래스 모델

- 객체의 구조와 행위 작성
  - 속성(Attribute)
    - 엔티티 클래스의 속성들은 대부분 스토리보드를 통해서 찾아지며 용어집에 작성되기 때문에 용어집을 보고 작성
    - 클래스의 연산 수행을 위해 특별히 요구되는 속성들은 유스케이스의 사건 흐름을 통해서 찾음
    - 처음에는 한글로 작성하고 점진적으로 영문으로 전환하여 설계 시점에서 나머지 용어들을 영문 전환
      - 분석 시 용어에 대한 정의와 속성에 대한 제약 사항들을 찾는 것이 더욱 중요하기 때문
  - 연관 클래스(Association Class)
    - 2개의 객체에서 다루어져야 할 정보가 있을 경우의 정보 표현
    - 다른 클래스처럼 행동할 수 있기 때문에, 관계를 가질 수 있음

- 객체의 행위 분석
  - UML에서 개개의 객체의 행위
    - 상태 다이어그램(State diagram)에서 표현
    - 상태 다이어그램은 많은 하위 객체를 포함하는 큰 클래스 또는 컨트롤 클래스의 행위 조사 시 유용
  - 상태 다이어그램



분석 클래스 모델

- 객체의 행위 분석
  - 상태(States)
    - 어떤 조건들을 만족시키거나, 행위를 수행하거나, 사건을 기다리는 동안 객체 수명주기(Life time)의 특정 시점에서의 상황
    - 한 객체의 상태는 클래스에 포함된 하나 또는 그 이상의 속성값에 의해 특성화
    - 상태 전이를 수행하는 행위 : → 진입(Entry) 행위로 작성
    - 상태 밖에서 모든 상태 전이를 수행하는 행위 → 진출(Exit) 행위
    - 상태 안에서 발생되는 행위 → 활동

분석 클래스 모델

- 객체의 행위 분석
  - 상태 전이(State Transitions)
    - 이전 상태에서 다음 상태로 바뀌는 것
    - 재귀적 상태 전이라면 원래 상태와 동일할 수 있음
    - 행위가 따를 수 있음
    - 자동 상태 전이
      - 원래 상태의 활동이 완료되었을 때
      - 일어난 전이와 연관된 사건의 이름 없음
    - 자동적이지 않은 전이
      - 이름을 갖는 사건에 의해 발생
      - 이 사건은 다른 객체 또는 시스템 외부에서부터 전달
    - 전이
      - 상태와 연관된 하나의 행위 또는 안내 조건 가짐
      - 사건 발생시킴
    - 행위
      - 상태 전이가 발생될 때 발생하는 행위

분석 클래스 모델

- 객체의 행위 분석
  - 특이 상태(Special States)
    - 상태 다이어그램에 추가되는 특이 상태는 2개
      - 시작 상태
      - 종료 상태
    - 모든 다이어그램은 하나의 시작 상태 가짐

분석 모델 검토

- 유스케이스 모델 검토
  - 액터, 유스케이스, 유스케이스 다이어그램, 유스케이스 명세서, 용어집, 스토리보드, 보조 명세서(Supplementary Specification)로 구성
  - 액터에 대한 주요 검토사항
    ① 액터에 대한 개요에 이해당사자가 시스템을 사용하는 경우를 반영하였는지 검토
    ② 액터가 하나의 역할을 수행하는지 검토
      - 만일 액터가 두 개 이상의 역할을 수행한다면 분리
    ③ 액터의 이름이 직관적이고 설명이 충분한지 검토
      - 사용자들과 고객들이 액터의 이름을 이해할 수 없다면 이름을 다시 명명

분석 모델 검토

- 유스케이스 모델 검토
  - 유스케이스에 대한 주요 검토사항
    - ① 유스케이스가 액터에게 가치(Value)를 제공하는 단위의 크기를 가지고 있는지 검토
    - ② 유스케이스 이름이 액터의 목적을 반영하고 있는지 검토
    - ③ 유사한 사건 흐름을 갖는 유스케이스들이 있는지 검토
      - 공통의 사건 흐름을 갖는다면, <<include>>로 유스케이스 구조화
      - 특정 조건에서만 다른 사건 흐름을 갖는다면, <<extend>>로 유스케이스 구조화

분석 모델 검토

- 유스케이스 명세서 검토
  - 유스케이스 실현을 작성하기 전에 유스케이스 사건 흐름에 대한 일관성 검토 수행
  - 유스케이스 사건 흐름 작성에 대한 일관성 유지를 위해서 용어집(Glossary)을 작성하고, 용어집에 정의된 용어 사용
  - 유스케이스 명세서의 주요 검토사항

    ① 기술 문장은

    - "액터가 … 하고 시스템에게 …를 요청한다"
    - "시스템은 …를 처리하고, 액터에게 …를 제공한다."
    - 액터의 책임과 목적 요청, 시스템의 권한과 책임으로 구성

    ② 사용자 수준에서 한 액터가 가치를 얻기까지의 크기에서 다른 액터(Secondary actor)가 처리해야 하는 일이 있을 때, 이 부분에 대한 기술은 액터의 처리 설명

    - "(액터이름)가 …를 처리한다."와 같이 기술

- 유스케이스 명세서 검토
  - 유스케이스 명세서의 주요 검토사항 (계속)

③ 어떠한 사건(Event)이 발생되었을 때, 처리해야 하는 부분에 대해서 “[사건 명]” 을 작성하고, 이때 수행해야 하는 흐름 작성

④ 사건 흐름은 메인 흐름, 대안 흐름, 예외 흐름으로 구성

- 유스케이스와 관련 된 비기능적 요구사항이나 설계 제약 사항들을 여러 차례 반복해서 찾음
- 예외 흐름은 액터의 행위에서의 예외와 시스템의 처리에서의 예외로 분리하여 작성
- 액터의 행위에 대한 예외의 책임은 분석 클래스의 경계 클래스가, 시스템의 처리에서의 예외는 컨트롤 클래스가 책임

⑤ 선행조건(pre-condition)은 유스케이스가 시작하는 조건 기술

- “액터가 …를 요청했을 때 시작한다.” 와 같이 기술

⑨ 후행조건(post-condition)은 유스케이스의 처리 결과 기술

- 유스케이스 명세서 검토
  - 개발자 수준의 유스케이스를 대상으로 한 유스케이스 기술서의 주요 검토사항

① 사용자 수준 유스케이스 기술을 사용자와 액터의 상호작용에서 사용자, UI, 관리자의 상호작용으로 정제

② 비즈니스를 처리하는 작업자와 작업자가 사용하는 엔티티와 처리를 위한 비즈니스 규정을 중심으로 작성

③ 관리자의 이름은 “유스케이스 이름 처리자 + 처리자”의 형태

- 작업자는 개발자 수준 유스케이스 각각에 대해서 하나씩 만듦

④ 엔티티는 개발자 수준 유스케이스 수행을 위해서 입력되는 정보, 유스케이스에 의해 생성되는 정보 또는 유스케이스 처리에 필요한 정보

⑤ 2차 액터가 시스템인 경우에는 엔티티 없이 직접 컨트롤에게 서비스를 요청하도록 작성

분석 모델 검토

- 분석 클래스와 분석 유스케이스 실현을 포함한 분석 패키지 구성
- 분석 유스케이스 실현에 대한 주요 검토사항

① 모든 예외적인 경우를 포함하여 메인, 서브 흐름들이 모두 다루어졌는지 검토

② 유스케이스의 행위들이 올바른 객체의 책임으로 분산되었는지 검토

- 분석 클래스 검토 : 분석 클래스들은 클래스의 종류에 따라 각자의 역할을 수행
  - » 경계 클래스 : 사용자와의 상호작용 담당
  - » 컨트롤 클래스 : 유스케이스 흐름에 대한 제어와 비즈니스와 관련된 논리 처리
  - » 엔티티 클래스 : 정보나 어떤 현상이나 개념, 실제 존재하는 객체나 사건에 대한 모델 역할
  - » 클래스들이 자신의 역할을 잘 수행하고 있는지 검토

분석 모델 검토

- 분석 유스케이스 실현에 대한 주요 검토사항
  - ③ 유스케이스의 행위들이 올바른 객체의 책임으로 분산되었는지 검토
    - 클래스 분할
      - » 여러 역할을 수행하는 클래스는 분할
    - 클래스 제거
      - » 클래스가 어떠한 구조와 속성 또는 행위를 가지지 않는 것이나, 어떠한 유스케이스에도 참여하지 않는 클래스가 있다면 이런 클래스들은 제거
      - » 컨트롤 클래스는 인간사의 중재자와 같은 역할
    - 클래스 관계
      - » 시퀀스 다이어그램이나 컴뮤니케이션 다이어그램에서 나타나는 메시지 전송
      - » 수신하는 객체들 사이에는 관계가 존재
      - » 송신 클래스와 수신 클래스 사이에 연관이나 집합의 관계가 작성되었는지를 검토하고, 만약 관계가 빠져 있다면 관계 추가

- 유스케이스 명세서
  - 분석 단계에서 해야 할 일 : 구성 요소들의 수행 업무 정리
    - ① 구성 요소들이 수행하기 위한 처리 업무
    - ② 일이 수행될 때 어떤 상태를 만족해야 하는지 기술
    - ③ 명세서 작성
  - 강좌개설 유스케이스 명세서

강좌개설 사용자 수준 유스케이스 사건 흐름
1. 액터는 승인정보**(id, password)**를 입력하고 시스템에게 승인을 요청한다. **2.** 시스템은 액터가 입력한 승인정보에 대한 유효성을 평가하고 승인을 처리하고 개설과목 리스트와 액터가 개설을 요청한 개설강좌 리스트를 제공한다 **[강좌개설요청 시]** **1.** 액터는 개설과목 리스트에서 강좌개설을 요청할 과목을 선택하고 시스템에게 강좌개설을 요청한다. **2.** 시스템은 요청한 개설과목을 강좌개설요청과목 리스트에 추가하고, 액터에게 갱신된 강좌개설요청 리스트를 제공한다. **[강좌개설요청 취소 시]** **1.** 액터는 강좌요청개설 리스트에서 취소할 강좌를 선택하고, 시스템에게 강좌요청개설 취소를 요청 한다. **2.** 시스템은 요청한 과목을 강좌개설요청과목 리스트에서 삭제하고, 액터에게 갱신된 강좌개설요청 리스트를 제공한다. **[시스템 강좌개설 마감일이 되었을 시]** **1.** 시스템은 학사정보를 갱신(강좌개설마감)으로 종합시간표를 출력 학생들에게 **E-Mail**로 전송한다.

- 분석 클래스 찾기
  - 엔티티 클래스
    - 액터에 의한 입력과 시스템이 액터에게 제공하는 출력이나, 시스템의 책임(시스템이 해야 하는 일이 무엇인가)을 조사함으로써 찾을 수 있음
    - 업무를 처리하는 개발자 수준 유스케이스에 존재
- 유스케이스 기술서 정련
  - 분석 클래스를 찾은 후, 유스케이스 기술서 정련
  - 정련된 사용자 수준 유스케이스의 사건 흐름의 형태
    - 액터가 경계 클래스에 입력사항이 있으면 입력을 하고, 서비스 요청
    - 경계 클래스가 사용자수준 컨트롤 클래스에게 서비스 요청
    - 사용자수준 컨트롤 클래스가 개발자수준 컨트롤 클래스에게 서비스 요청
    - 경계를 구성하기 위한 정보들은 컨트롤 클래스가 직접 엔티티에게 요청

- 유스케이스 기술서 정련(계속)

강좌개설 사용자 수준 유스케이스 사건 흐름
[사용자 승인 시] 1. 액터는 강좌개설UI에 승인정보(id, password)를 입력하고 승인을 요청한다. 2. 강좌개설UI는 강좌개설관리자에게 사용자 승인을 요청한다. [사용자 승인 include] 3. 강좌개설관리자는 사용자승인작업자에게 승인을 요청한다. 4. 강좌개설관리자는 승인정보가 유효하면, 4.1 개설과목에게 개설과목리스트를 요청한다. 4.2 개설강좌에게 개설강좌요청리스트를 요청한다. 5. 강좌개설관리자는 강좌개설UI에게 개설과목리스트와 개설강좌요청리스트를 보여주기를 요청한다. [강좌개설요청 시] 1. 액터는 강좌개설 UI의 개설과목 리스트에서 강좌개설을 요청할 과목을 선택하고, 강좌개설 UI에게 강좌개설을 요청한다. 2. 강좌개설관리자는 강좌개설 UI에게 강좌개설을 요청한다. [강좌개설처리 include] 3. 강좌개설관리자는 강좌개설처리작업자에게 강좌개설을 요청한다. [강좌개설요청 취소 시] 1. 액터는 강좌개설 UI의 개설강좌 리스트에서 취소할 강좌를 선택하고, 강좌개설 UI에게 강좌개설요청 취소를 요청한다. 2. 강좌개설 UI는 강좌개설관리자에게 강좌개설요청 취소를 요청한다. [강좌개설처리 include] 3. 강좌개설관리자는 강좌개설처리관리자에게 강좌개설요청 취소를 요청한다. [시스템 - 강좌개설 마감일이 되었을 시] [강좌개설마감처리 include] 1. 시스템은 강좌개설마감처리 관리자에게 학사정보를 갱신(강좌개설마감)으로, 종합시간표를 출력, 학생들에게 E-Mail로 전송을 요청한다.

- 유스케이스 기술서 정련 (계속)

Use case : 강좌개설처리 업무

"이 유스케이스는 강좌개설처리작업자가 교수의 강좌개설요청이나 강좌개설요청 취소를 수행하는 업무이다."

1. 작업자 명 : 강좌개설처리작업자

2. 강좌개설요청

2.1 관련정보

2.1.1 입력정보 : 교수식별자, 개설과목식별자, 학기

2.1.2 생성정보 : 개설강좌

2.2 선행조건

교수의 개설강좌리스트에 요청한 과목이 존재하지 않아야 한다.

2.3 후행조건

교수의 개설강좌리스트에 요청한 과목이 추가된다.

3. 사건 흐름

a. 강좌개설처리작업자는 개설과목에게 개설강좌추가를 요청한다.

b. 개설과목은 개설강좌에게 추가를 요청한다.

한국 IT 대학 분석 모델

- 클래스의 문서화
  - 분석 클래스가 만들어질 때마다 유스케이스처럼 문서화
  - 문서화는 클래스의 구조에 대해서가 아니라 클래스 목적 기술
  - 잘된 문서화의 예
    - “학생들을 등록하고 등록금을 지불하도록 하는 데 요구되는 정보로써, 학생은 대학에서 과목을 선택하고 등록되는 사람을 의미한다.”
  - 잘못된 문서화의 예
    - “학생의 이름, 주소, 전화번호”
    - 이러한 정의는 단지 클래스의 구조만을 말하고 있어서, 왜 이 클래스가 필요한지에 대한 내용이 없음
  - 클래스 이름을 정하고 문서화하는데 어렵다는 것은 좋은 추상화가 아니라는 것!

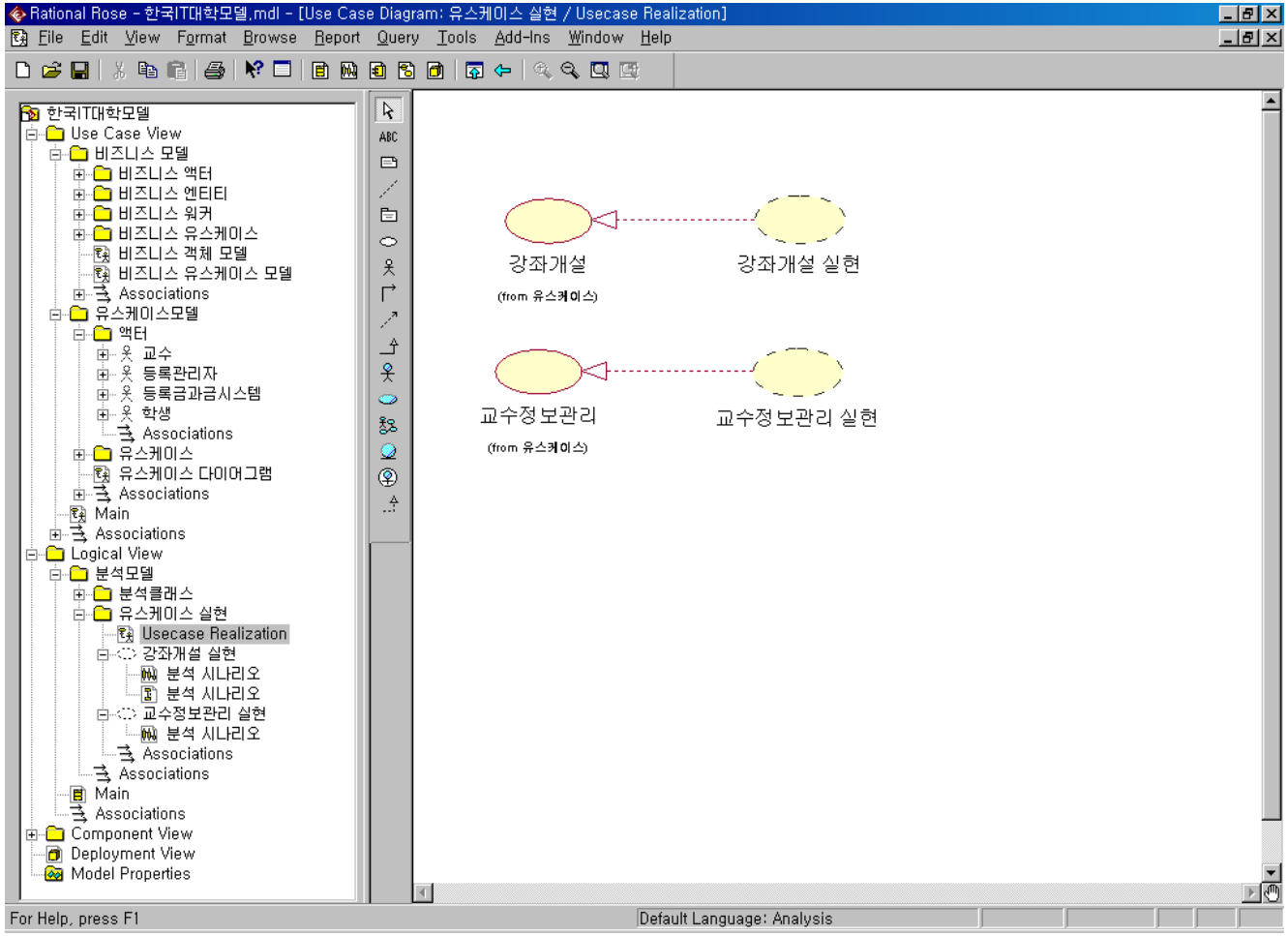
- 클래스의 문서화 (계속)
  - 클래스의 이름을 정하고 문서화할 때 발생하는 것
    - 이름을 식별할 수 있고 분명하고 간결하게 정의할 수 있다면 좋은 후보 클래스
    - 이름은 식별할 수 있지만 다른 클래스와 정의가 같다면 클래스를 조합
    - 이름은 식별할 수 있지만 클래스 목적을 문서화하는데 책 1권 정도의 분량이 필요하다면 이것은 너무 큰 클래스이므로 분할
    - 이름을 식별하기 어렵고 정의하기도 어렵다면 올바른 추상화를 결정할 필요가 있으므로 좀 더 분석

분석 클래스 다이어그램 작성

- 클래스 다이어그램은 분석 모델의 클래스에 대한 그림이나 뷰를 제공을 위해 만듦
- 분석 모델의 논리 뷰(Logical View)[6]의 메인 클래스 다이어그램은 전형적으로 시스템의 분석 패키지들에 대한 그림
- 각각의 패키지
  - 자신의 메인 클래스 다이어그램을 가질 수 있음
  - 패키지의 전역(Public) 클래스를 위치시키고 다른 다이어그램들은 필요에 의해 만들어짐

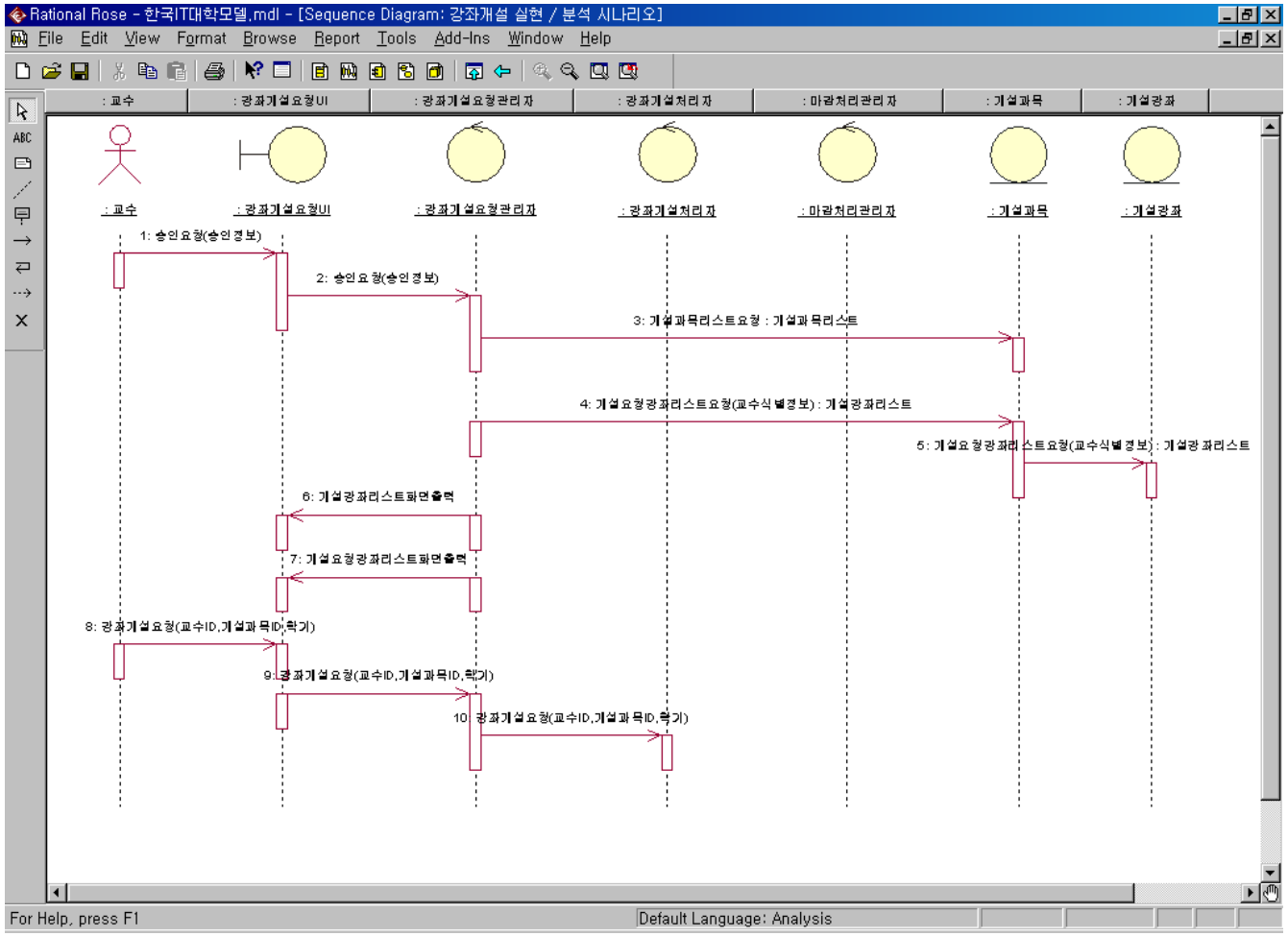
한국 IT 대학 분석 모델

- 한국IT대학 분석 모델 다이어그램 예 (starUML로 실습!)
  - 유스케이스 실현을 위한 컴뮤니케이션 생성



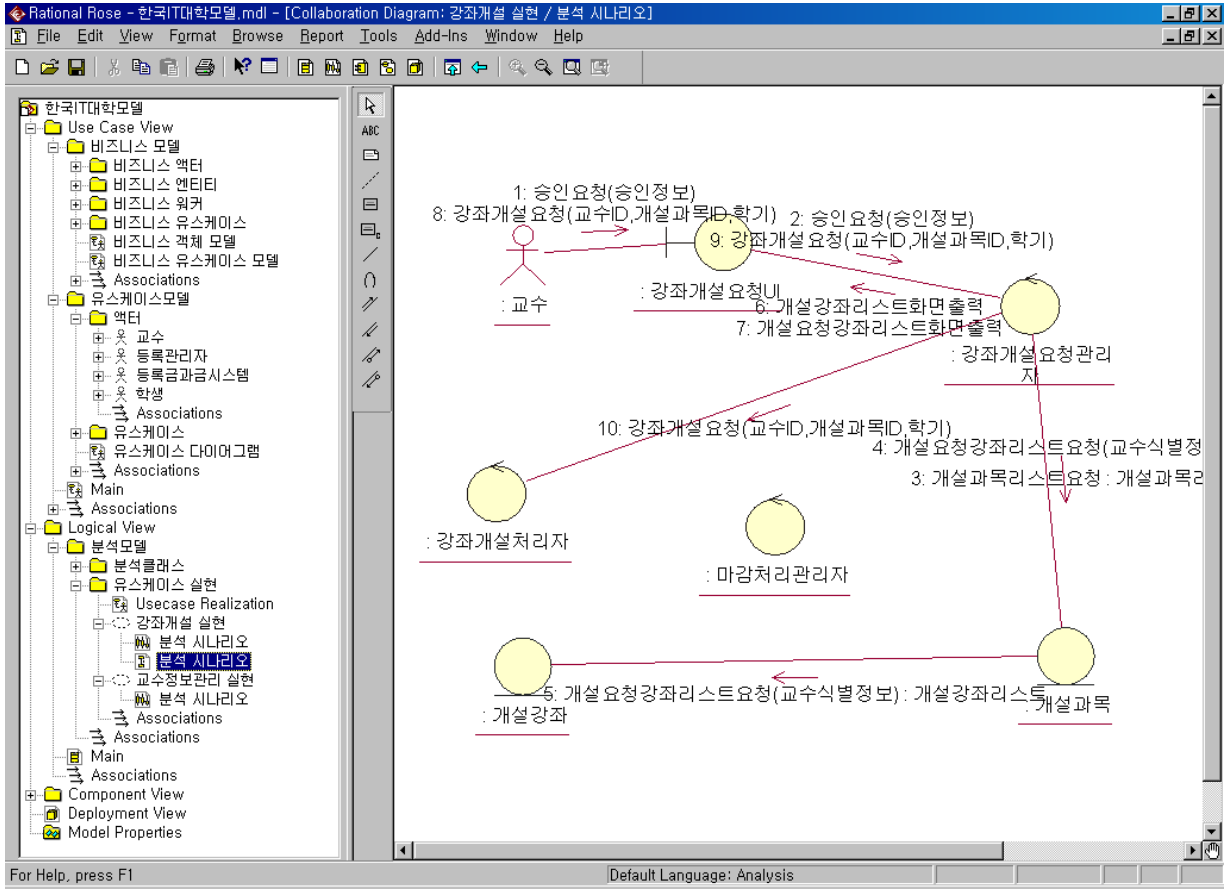
한국 IT 대학 분석 모델

- 한국IT대학 분석 모델 다이어그램 예
  - 강좌개설 실현 컴유니케이션의 시나리오(시퀀스 다이어그램)



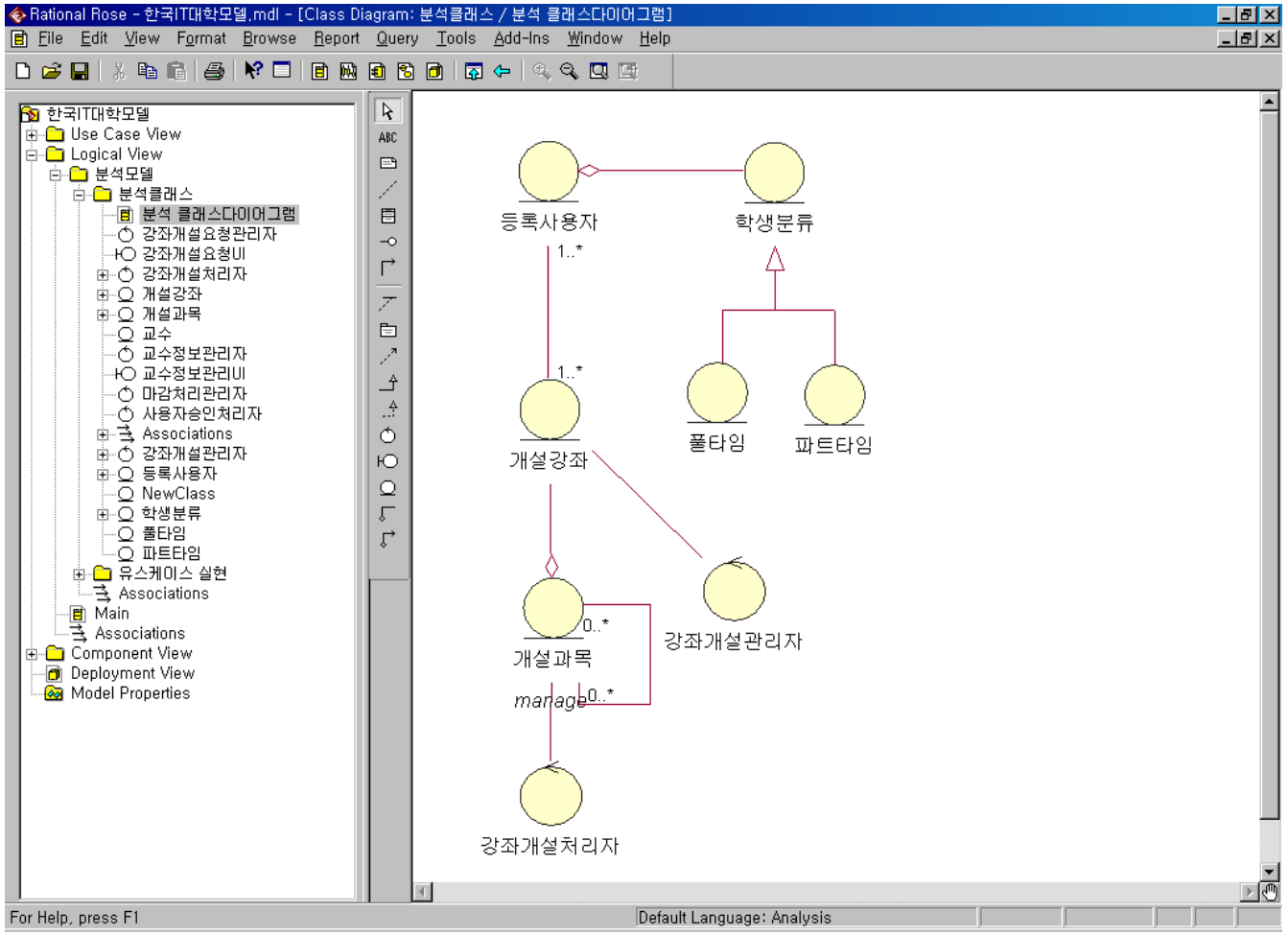
한국 IT 대학 분석 모델

- 한국IT대학 분석 모델 다이어그램 예
  - 강좌개설 실현 컴뮤니케이션의 시나리오(컴뮤니케이션 다이어그램)



한국 IT 대학 분석 모델

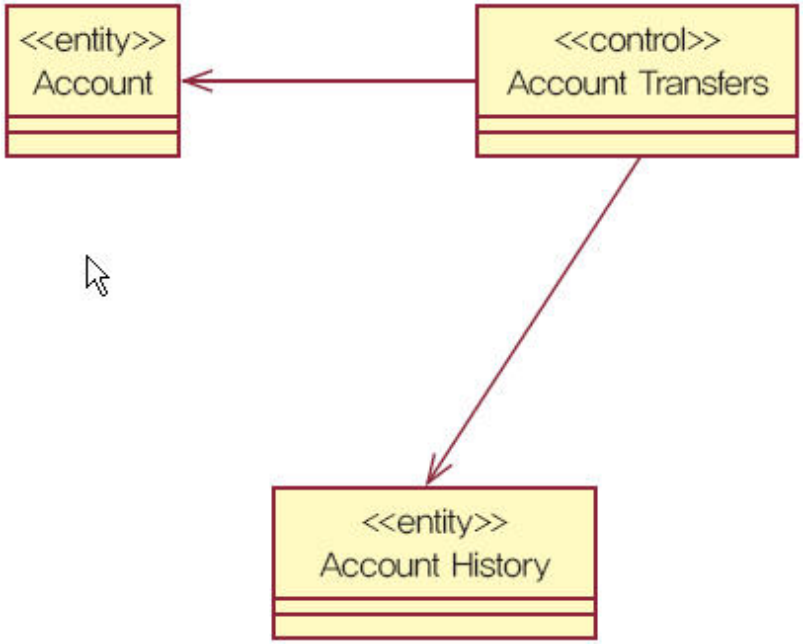
- 한국IT대학 분석 모델 다이어그램 예
  - 분석 클래스 다이어그램



산출물 사례 및 작성 지침

- 분석 클래스 명세서

분석클래스다이어그램		작업흐름		A00 분석 모델리			
시스템명		작성자		작성일자			
		문서번호	AN010	버전		쪽	



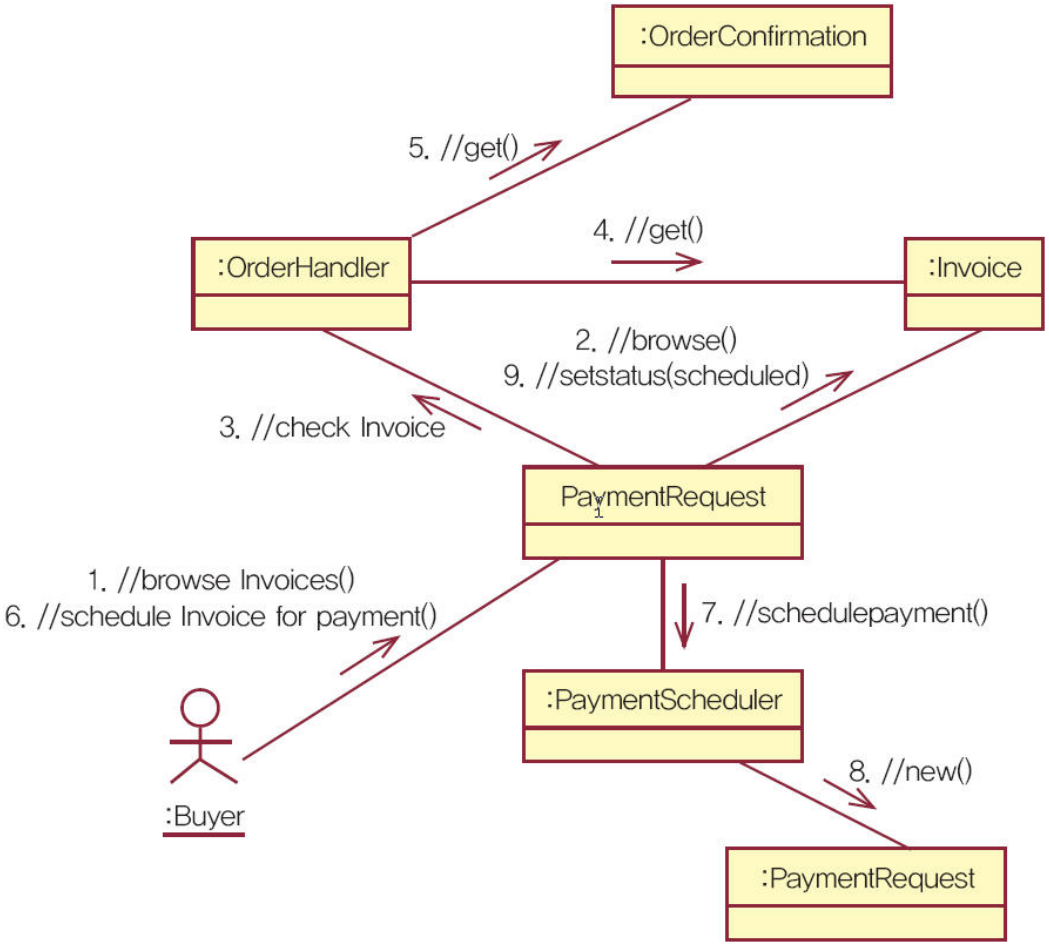
- 분석 클래스 명세서
  - 작성지침
    - 개발 시스템을 기능 요구사항과 문제 영역에 기반하여 여러 부분으로 구분(패키지)하여 개발하게 되는데, 각 부분을 구성하는 구성 요소들과 그들 간의 관계를 보여주기 위한 목적으로 클래스 다이어그램 작성
    - 분석 클래스 다이어그램에는 시스템을 서브시스템과 패키지로 구성하였을 때 가장 하위단에 해당하는 패키지나 서브시스템을 구성하는 클래스들 간의 관계 설명
  - 구분
    - 구분란에는 해당 클래스 다이어그램이 소속된 경로 기술

- 분석 클래스 명세서

유스케이스 실현		작업흐름		A00 분석 모델리			
시스템명		작성자		작성일자			
		문서번호	AN020	버전		쪽	

유스케이스 명	Billing & payment system / pay invoice
특별 요구사항	**buyer**가 발부받은 송장 검색 요청 시, 화면에 송장을 가져오는 데 소요되는 시간은 **0.5**초 미만이어야 한다. 송장 지불은 **SET** 표준을 사용하여야 한다. **Invoice** 클래스는 지속성 클래스이다. **OrderHandler** 클래스는 시간당 **10,000**건의 트랜잭션을 처리할 수 있어야 한다.

- 분석 클래스 명세서
  - 컴유니케이션 다이어그램



산출물 사례 및 작성 지침

- 분석 클래스 명세서
  - 작성지침
    - 분석 단계에서의 유스케이스 실현은 요구사항 분석에서 식별된 시스템의 기능 요구사항(시스템 행위)을 시스템 내부 관점에서 좀 더 상세 파악 목적
    - 분석 클래스의 책임으로 할당
      - 시스템 외부 관점에서 파악된 요구사항이 시스템 내부의 자원에 분배되어 각 자원이 담당해야 할 책임 결정
      - 해당 유스케이스의 사건 흐름을 실현하는 데 필요한 분석 클래스를 식별
      - 해당 유스케이스에 기술된 시스템 행위
  - 유스케이스명
    - 유스케이스 모델의 유스케이스 명과 동일한 유스케이스 명 기술

산출물 사례 및 작성 지침

- 분석 클래스 명세서
  - 특별 요구사항
    - 설계, 구현 단계에서 기술적으로 해결하는 비기능적 요구사항 기술
  - 클래스 다이어그램
    - 해당 유스케이스 실현에 참여하는 분석 클래스와 그들 간의 정적 관계를 설명하는 클래스 다이어그램 작성
  - 시퀀스/컴뮤니케이션 다이어그램
    - 해당 유스케이스의 행위를 실현하는 인터액션 다이어그램 작성
  - 구분
    - 기본 흐름인지 선택 흐름인지 구분
    - 선택 흐름의 경우 해당 흐름명 기술
    - 기본 흐름에도 흐름명이 부여될 수 있는 경우, 해당 흐름 명 기술

교재#2의 6장 요약

- 객체지향 기법을 적용한 개발 프로젝트에서 식별된 요구사항에 대한 분석 방안 수립 및 실행능력 배양
- 팀 프로젝트에서 실행방법 모색 및 적용
- 시스템 개발을 위한 요구사항 분석의 산출물 작성 및 검토방법 이해
- 관련 개념 정리

강의 계획 피드백(9주차)

주차	강의주제	강의내용	과제	평가
1주차	객체지향 패러다임	과목 소개 및 객체지향 방법론의 전반적인 개요		
2주차	프로젝트 관리1	프로젝트 계획 및 팀 편성/프로젝트 과제 제시		
3주차	소프트웨어 개발방법론과 UML	기존의 소프트웨어 개발방법론과 객체지향방법론 차이점 이해	과제1 : 프로젝트 현장 및 계획서 제출(5)	
4주차	Use Case와 UML	UML 특성 이해		
5주차	UP(Unified Process) 방법론	UP 방법론 이해		
6주차	비즈니스 모델링 및 요구사항 정의	사례를 통한 비즈니스 모델링 및 요구사항 정의 방법 이해	과제2 : 요구사항 정의 결과 제출(5)	
7주차	분석 모델링 및 UML 다이어그램(분석)	객체지향 분석 방법 이해 및 분석용 UML 다이어그램 작성 방법 이해		중간고사
8주차	분석 결과 문서화 및 설계 모델링	분석 산출물 작성 방법 및 객체지향 설계 방법 이해	과제3 : 분석 결과 제출(10)	
9주차	UML 다이어그램(설계)	설계용 UML 다이어그램 작성 방법 이해		
10주차	객체 설계	객체설계 및 세분화		
11주차	설계 결과의 문서화 및 프로젝트 관리 2	시스템 설계 결과의 문서화 방법 이해 및 형상관리/검증과 확인 방법 이해	과제4 : 설계 결과 제출(10)	
12주차	시스템 구현	객체지향 프로그래밍의 기본 개념 및 기법		
13주차	시스템 테스트 및 구현/시험 결과의 문서화	객체지향 테스팅 기법 및 구현/시험 산출물의 문서화 방법 이해	과제5 : 구현/시험 결과 및 유지보수 계획 제출(20)	
14주차	프로젝트 관리3	소프트웨어 품질관리와 프로세스 개선 방법 이해		
15주차	최종 결과 문서화 및 발표	최종 산출물 문서화 방법 이해 및 개발 결과 발표	과제6 : 최종보고서 제출 및 발표(10)	

다음 주(10주차) 강의계획

- 교재#2의 6장 완전하게 이해하기
- 교재#1의 14장, 교재#2의 7장 읽어오기
- 팀 프로젝트 진행 : 계획단계-분석단계-설계단계
  - 프로젝트 헌장(PC) : 완료(2016.09.28, 18:00)
  - 프로젝트 관리 계획서(PMP) : 완료(2016.10.12.18:00)
  - 소프트웨어 요구사항 정의서(SRD) : 완료(2016.10.12.18:00)
  - 소프트웨어 요구사항 명세서(SRS)
    - 1차 제출(2016.10. 30, 24:00)
    - 최종 제출(2016.11.09, 18:00)

☞ 다음 시간(9주차-2)은 **중간고사**
다음 주(10주차)는 **교재#1의 14장, 교재#2의 7장 강의**