노동부 Ministry of Labor 중소기업직업훈련컨소시엄

Korea Software Technology Association

UML2.x 기초 다루기

훈련기간: 2010.01.25 ~ 02.05

강사명: 손재현 *-넥스트리소프트*
-jhsohn@nextree.co.kr

과정 개요

❑ 교육 목표 & 특징

- UML2.x의 이해
- 유스케이스 작성
- 객체모델링 이해
- UML2.x의 다양한 다이어그램 이해 및 활용
- 모델링 도구 사용법 습득

강의 요구 기술

- **본 강의는 아래 기술에 대한 이해를 필요로 합니다.**
  - 객체지향 언어(Java) 기초
  - 개발프로세스 이해

교육 일정표

❑ 교육은 매 회 4 시간씩 총 5회에 걸쳐 진행합니다.

1 일차	2 일차	3 일차	4 일차	5 일차
- UML 개요 UML 소개 UML 역사 UML 다이어그램분류	- 구조 다이어그램 클래스 객체 컴포넌트 배치	- 행위 다이어그램 유스케이스 액티비티 상태기계	- 상호작용 다이어그램 상호작용 Overview 시퀀스 커뮤니케이션 타이밍	- 유스케이스 I 유스케이스 개요 유스케이스 내용 유스케이스 다이어그램
6 일차	**7 일차**	**8 일차**	**9 일차**	**10 일차**
- 유스케이스 II 유스케이스 목표수준 유스케이스 명세 유스케이스 패턴	- 유스케이스 III 유스케이스 분석기법 분석클래스 제어클래스 실체클래스	- 요구사항 모델실습 I 유스케이스 사용자 시나리오 핵심개념 모델	- 요구사항 모델실습 II 인터페이스 추출 유스케이스 분석 컴포넌트 식별	- 설계모델 실습 컴포넌트 설계 유스케이스 설계 도메인 모델

7일차 – 유스케이스 분석 기법

1 개요
2 경계클래스(Boundary Class)
3 제어클래스(Control Class)
4 실체클래스(Entity Class)
5 연관 관계

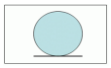
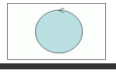
유스케이스 분석 기법 개요

❑ 개요

- 객체지향 및 컴포넌트 기반 개발방법론에서 클래스를 도출하는 체계적인 기법으로 사용
- 야곱슨에 의해서 제안된 방법으로 요구사항 정의 활동에서 도출된 유스케이스를 분석함으로써 세 가지 종류의 클래스(경계 클래스, 제어 클래스, 실체 클래스)를 파악함
- 분석 활동에서 파악되는 경계, 제어, 실체 클래스를 설계 활동에서의 클래스와 구분하기 위해서 분석 클래스(analysis class)라고 부름

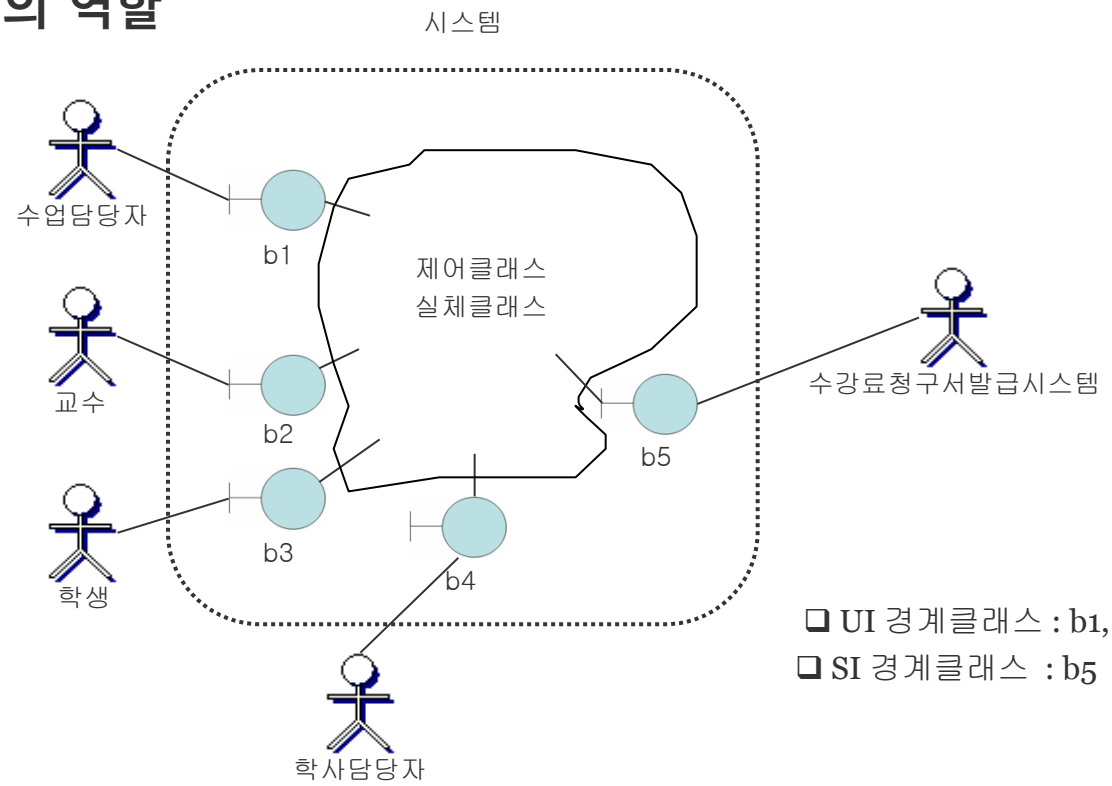
분석 클래스 종류

- **경계 클래스(boundary class)**
  - 개발될 시스템과 그 외부와의 연결을 담당하는 클래스
  - 사용자 인터페이스화면 및 외수 시스템과 통신하는 클래스
- **제어 클래스(control class)**
  - 시스템이 제공하는 비즈니스 로직을 구현하는 클래스
  - 유스케이스로 정의된 비즈니스 로직을 다른 클래스들의 개체를 이용하면서 제공하는 역할
- **실체 클래스(entity class)**
  - 영속성이 있는 데이터에 대한 관리 기능을 제공
  - 즉, 영속성이 있는 데이터를 생성, 조회, 수정, 삭제하는 기능

종류	스테레오 타입	아이콘	역할
실체클래스	<<entity>>		영속성이 있는 데이터에 대한 조작을 담당
경계클래스	<<boundary>>		시스템과 외부와의 연결을 담당
제어클래스	<<control>>		비즈니스 로직을 담당

경계클래스 개요(1/2)

- **경계클래스(Boundary Class)는 개발될 시스템과 그 외부와의 연결을 담당**
  - 시스템의 가장 외곽에 존재하여 시스템의 다른 클래스들을 대신해서 시스템 외부와의 상호작용을 전담하는 클래스
  - 시스템의 외부는 액터를 의미
    - 시스템 외부에 존재하는 액터와의 상호작용을 제공하는 역할
- **경계클래스의 역할**



- UI 경계클래스 : b1, b2, b3, b4
- SI 경계클래스 : b5

경계클래스 개요(2/2)

- 제어클래스와 실체클래스는 시스템 외부의 액터와 직접 연결되지 않음
- 경계클래스가 시스템 내에 존재하는 많은 클래스 중에서 시스템 외부의 존재 즉 사용자 및 외부시스템을 연결하는 역할
  - 시스템 내의 다른 모든 클래스(제어클래스와 실체클래스)는 사용자 또는 외부 시스템과 직접 연결되지 못하며 반드시 경계클래스를 통해서 사용자 또는 외부 시스템과 연결
- UI(User Interface) 경계클래스
  - 사용자 액터와의 연결을 담당하는 경계클래스
- SI(System Interface) 경계클래스
  - 외부 시스템 액터와의 연결을 담당하는 경계클래스

UI 경계클래스(1/8)

- 시스템의 사용자(사용자 액터)는 사용자 인터페이스 화면을 통해서 시스템이 제공하는 기능을 이용
- 사용자가 시스템을 사용하기 위해서 보는 각각의 화면(또는 윈도우)은 모두 경계클래스에 해당
  - 이러한 종류의 경계클래스를 UI 경계클래스(User interface boundary class)
  - UI 경계클래스는 사용자 인터페이스를 위한 각 화면을 뜻하는 개념적인 용어
- UI 경계클래스의 실질적 의미

개발 언어	UI 경계클래스의 대응 요소
Visual Basic	Form
JSP	JSP 페이지
ASP.NET	ASP.NET 페이지

개발 언어	UI 경계클래스의 대응 요소
VC++/MFC	Window
ASP	ASP 페이지

UI 경계클래스(2/8)

❑ UI 경계클래스의 도출

- UI경계클래스는 사용자화면
- 유스케이스명세서의 ~화면 하나하나가 UI경계클래스에 해당

기본 흐름

1. 사용자는 시스템에 접속한다.
2. 시스템은 로그인 화면을 보여준다.
3. 사용자는 아이디와 암호를 입력하고 로그인을 선택한다.
   사용자 아이디 즉, 학번/교수번호/직원번호는 다음과 같은 규칙을 가진다.
   - ■ 학번 : 첫 문자는 'S'로 시작하고 이어서 3자리의 숫자가 나온다.
   - ■ 교수번호 : 첫 문자는 'P'로 시작하고 이어서 3자리의 숫자가 나온다.
   - ■ 직원번호 : 학사담당자는 'H'로 시작하고 이어서 3자리의 숫자가 나온다. 수업담당자는 'G'로 시작하고 이어서 3자리의 숫자가 나온다.

   사용자 암호는 7자리의 영문자 및 숫자로 구성된다.
4. 시스템은 입력된 아이디와 암호의 유효성(A1)과 정확성(A2)을 확인하고 사용자 별 메인 화면을 보여준다. 각 사용자 별 메인 화면은 다음과 같다.
   - ■ 학생 : 학생 메인 화면
   - ■ 교수 : 교수 메인 화면
   - ■ 학사담당자 : 학사 관리 메인 화면
   - ■ 수업담당자 : 수업 관리 메인 화면

대안 흐름

(A1) 유효하지 않은 아이디 또는 암호인 경우

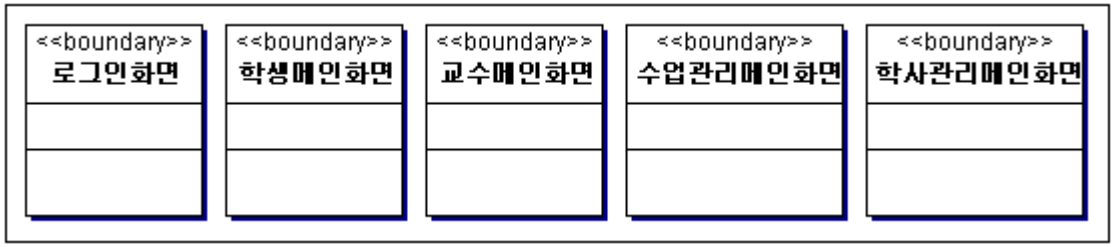
1. 시스템은 입력된 아이디 또는 암호가 허용되지 않는 값임을 알려준다.
2. 사용자는 다시 아이디와 암호를 입력함으로써 로그인을 시도한다.

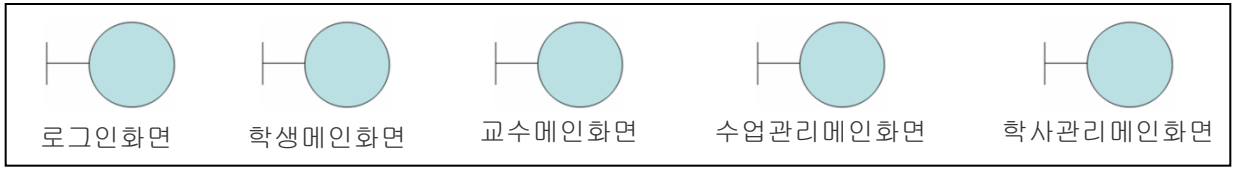
(A2) 부정확한 아이디와 암호인 경우

1. 시스템은 아이디와 암호가 부정확함을 알려준다.
2. 사용자는 다시 아이디와 암호를 입력함으로써 로그인을 시도한다.

UI 경계클래스(3/8)

❑ 로그인 유스케이스의 UI 경계클래스





UI 경계클래스(4/8)

❑ 암호변경 유스케이스의 UI 경계클래스

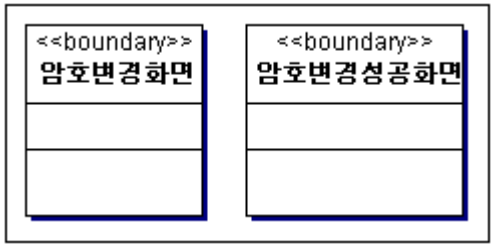
기본 흐름

1. 사용자는 사용자 별 메인 화면에서 암호변경을 선택한다.
2. 시스템은 암호변경 화면을 보여준다.
3. 사용자는 새 암호를 두 번 입력하고 변경을 선택한다.
4. 시스템은 사용자의 암호를 입력된 새 암호로 변경하고, 암호변경 성공 화면을 보여준다(A1).

대안 흐름

(A1) 입력된 두 암호가 일치하지 않는 경우

1. 시스템은 입력된 두 암호가 일치하지 않음을 알린다.
2. 사용자는 다시 새 암호를 입력해서 암호변경을 시도한다.



UI 경계클래스(5/8)

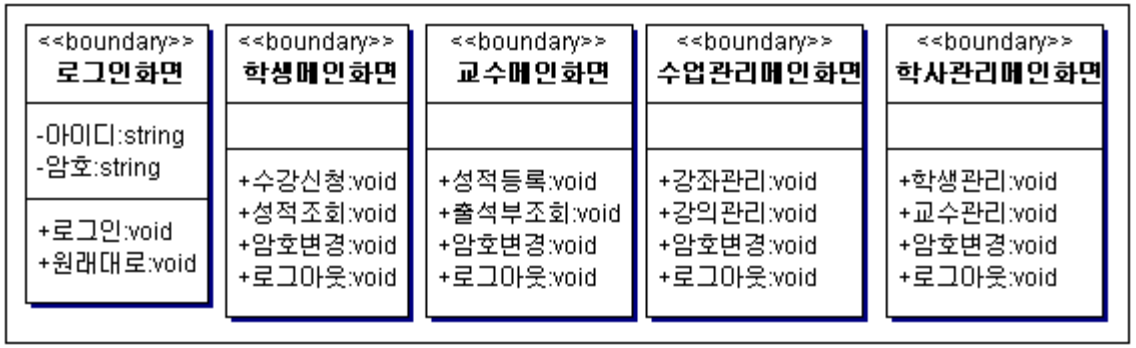
❑ UI 경계클래스의 속성

- 클래스의 속성은 클래스로부터 생성된 객체의 상태를 저장하는 공간으로, 클래스의 오퍼레이션들이 공유하는 변수
- UI 경계클래스는 사용자 화면에 해당하며, 각 화면마다 저장되어야 하는 상태는 사용자가 화면에 입력한 값 또는 시스템이 사용자에게 보여주기 위해서 출력한 값
- UI 경계클래스의 속성에는 화면을 통해서 입·출력되는 정보

UI 경계클래스(6/8)

❑ UI경계클래스의 오퍼레이션

- 클래스의 오퍼레이션은 클래스로부터 생성된 객체가 외부(또 다른 객체)에 제공할 수 있는 기능
- 사용자 화면이 제공하는 구체적인 기능의 수행은 화면 상의 메뉴 또는 버튼 등에 의해서 시작됨



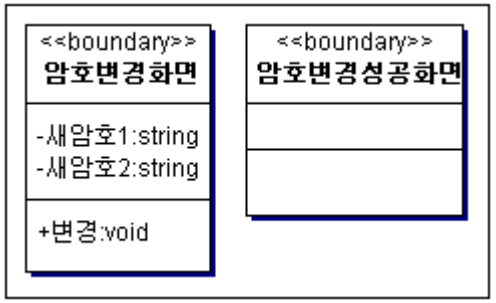
로그인 유스케이스의 경계클래스

- 오퍼레이션의 이름
  - 화면에서 보이는 버튼의 캡션 또는 메뉴 항목의 캡션을 이름으로 오퍼레이션을 정의
- 접근 권한
  - UI 경계클래스의 오퍼레이션은 외부(사용자)에 의해서 호출되는 것이므로 공용
- 오퍼레이션의 파라메터
  - UI 경계클래스의 오퍼레이션에는 아무런 파라메터가 없는 것이 정상
- 반환 타입
  - UI 경계클래스의 오퍼레이션의 반환 타입은 의미가 없으므로 생략

UI 경계클래스(7/8)

❑ UI경계클래스의 오퍼레이션

- 암호변경 성공 화면처럼 속성과 오퍼레이션이 하나도 없으면서 그 의미가 명확한 화면은 유스케이스명세서의 이벤트 흐름에서 간단히 출력할 메시지만 기술하고 별도의 화면을 정의하지 않는 것도 하나의 방법
- 암호변경 유스케이스의 경계클래스



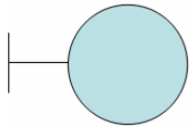
UI 경계클래스(8/8)

❑ UI 경계클래스의 도출 방법

도출 단위	유스케이스명세서의 이벤트 흐름에 기술된 각 화면마다 정의	
클래스 이름	유스케이스명세서의 이벤트 흐름에 명시된 화면 명에 해당	
속성	해당 화면에서 사용자가 입력하는 항목 및 시스템이 출력하는 항목에 해당	
	이름	입력/출력되는 데이터 항목의 이름에 해당
	접근 권한	전용
	스테레오 타입	❑ <<in>> : 입력 ❑ <<out>> : 출력 ❑ <<inout>> : 입출력
	타입	❑ **Boolean, String, Number, Date** 등 일반적인 것을 사용 ❑ 파라메터의 이름만으로 명확한 경우에는 생략 가능
오퍼레이션	해당 화면에서 시스템의 기능 수행을 시작시키기 위하여 선택되는 메뉴 항목 또는 각종 버튼에 해당	
	이름	화면에 표시되는 메뉴 항목의 캡션 또는 버튼의 캡션
	접근 권한	공용
	반환 타입	없음
	파라메터	없음

SI 경계클래스

- **SI 경계클래스는 경계클래스의 한 종류로써 개발할 시스템과 연동되는 외부시스템과의 연결을 담당하는 클래스**
  - 즉 유스케이스모델에서 파악된 시스템액터와의 연결을 담당하는 클래스
  - 일반적으로 SI 경계클래스는 정의된 프로토콜에 따라서 외부시스템에 데이터를 보내거나 받은 역할을 수행
- **SI 경계클래스의 도출 방법**
  - 외부 시스템 액터마다 하나의 SI 경계클래스를 정의
  - SI 경계클래스는 대응하는 외부 시스템 액터의 이름에 '에이전트'를 추가하여 이름을 줌으로써, SI 경계클래스가 외부의 어떤 시스템과의 연결을 담당하는지를 쉽게 파악할 수 있게 함
- **SI 경계클래스의 예**

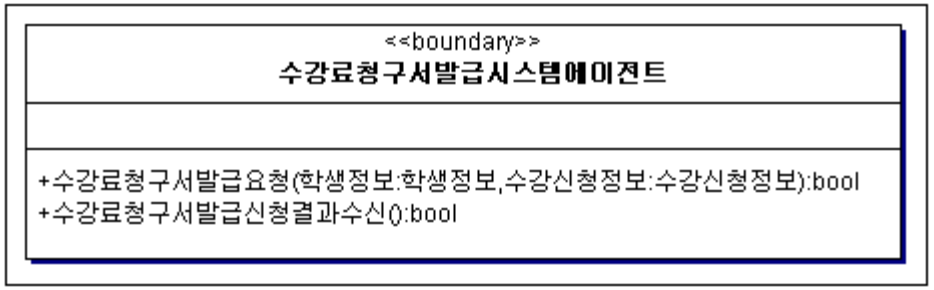


수강료청구서발급시스템에이전트

SI 경계클래스-오퍼레이션(1/3)

❑ SI 경계클래스의 오퍼레이션

- 외부 시스템과의 연결을 담당하므로 외부 시스템에 대한 기능의 수행 요구와 외부 시스템으로부터의 접속에 대한 처리가 오퍼레이션으로 정의
- 수강료청구서발급시스템에이전트 클래스의 오퍼레이션



- 수강료청구서발급결과수신
  - 외부 시스템의 기능 수행의 결과를 수신
- 외부 시스템의 기능 수행을 요청하는 오퍼레이션의 이름
  - '수행될 외부 시스템의 기능' + '요청'
- 요청 시 외부 시스템에 전송될 데이터
  - 학생정보, 수강신청정보처럼 오퍼레이션의 파라메터로 전달
- 외부 시스템에 대한 수행 요청 자체의 성공여부
  - Boolean 타입으로 반환

SI 경계클래스-오퍼레이션(2/3)

- **SI 경계클래스의 오퍼레이션은 외부 시스템과 접속을 해야 하기 때문에 비동기적으로 수행될 수 있음**
  - 네트워크를 통해서 외부 시스템에 데이터를 전달하면 그 결과를 항상 바로 얻을 수 있는 것이 아니며, 또는 외부 시스템의 수행 결과가 반드시 온다고 확신 못함
  - 비동기적으로 외부 시스템과 연동하는 경우에 외부 시스템은, 요청 받은 기능을 수행한 후에 그 결과를 시스템에 통신을 통하여 전달
    - 외부 시스템으로부터의 접속을 수신하고 이를 처리하는 기능도 오퍼레이션으로 포함되어야 함
    - 수강료청구서발급시스템에이전트의 수강료청구서발급결과수신() 오퍼레이션
  - 외부 시스템의 기능 수행의 결과를 수신하는 오퍼레이션의 이름
    - '수행된 외부 시스템의 기능' + '결과수신'
  - 수신 시 외부시스템에서 전송 받을 데이터
    - 네트워크를 통해서 전송 받은 데이터를 분석할 것이므로, 아무런 파라메터가 필요 없음
  - 외부 시스템에 대한 수행 결과 자체의 성공여부
    - Boolean 타입

SI 경계클래스-오퍼레이션(3/3)

- 만약, 외부 시스템과 비동기적인 방식이 아니라, RPC, RMI, .NET Remoting과 같은 동기적 통신을 한다면 수강료청구서발급결과수신()과 같은 오퍼레이션은 정의될 필요가 없음
- 구축할 시스템에서 외부 시스템에 전달할 데이터의 종류 및 형식과 수신할 결과 데이터의 종류 및 형식 등이 명확히 파악되고 정의되어야 함

SI 경계클래스-속성

- **SI 경계클래스의 속성은 두 가지 측면에서 정의**
  - 접속할 외부 시스템에 전달할 데이터를 저장하기 위해서 속성을 사용
  - 외부 시스템의 수행 결과를 수신하고 이 결과를 저장하기 위해서 속성을 사용
- **클래스의 여러 오퍼레이션들 사이에 공유하는 정보가 하나도 없는 경우에는 클래스에 속성이 정의될 필요가 없음**
- **SI 경계클래스의 경우에는 클래스의 오퍼레이션들은 각각 독립적인 기능을 수행하는 경우가 많으며, 이 오퍼레이션들이 공유하는 정보가 없는 것이 일반적**
  - '수강료청구서발급요청()' 오퍼레이션을 호출할 때 필요한 모든 데이터를 이 오퍼레이션의 파라메터로 넘겨주며, '수강료청구서발급요청()' 오퍼레이션과 '수강료청구서발급결과수신()' 오퍼레이션이 공유할 정보가 없는 경우가 많음
  - SI 경계클래스의 경우에는 속성이 정의되지 않은 경우가 빈번함

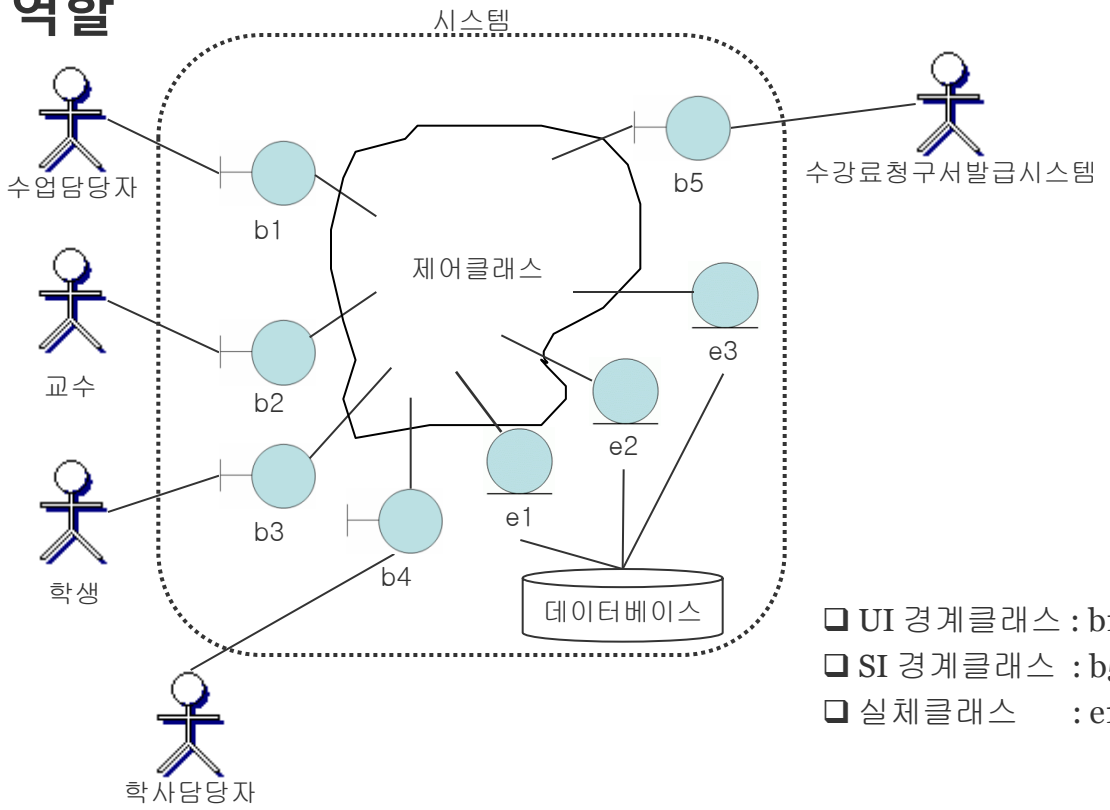
SI 경계클래스 요약

❑ SI 경계클래스의 도출 방법

도출 단위	유스케이스 모델의 각 시스템 액터마다 정의	
클래스 이름	시스템 액터 명 + '에이전트'	
속성	오퍼레이션들 사이에 공유 될 정보가 없는 경우가 많으므로 대부분의 경우에 속성이 없는 경우가 일반적이다.	
오퍼레이션	외부 시스템의 기능 수행을 요청하거나, 외부 시스템의 기능 수행 결과를 수신하는 역할	
	이름	❑ 외부시스템의 기능 + '요청' ❑ 외부시스템의 기능 + '결과수신'
	접근 권한	공용
	반환 타입	❑ '~요청' 오퍼레이션 : **Boolean** ❑ '~결과수신' 오퍼레이션 : **Boolean**
	파라메터	❑ '~요청' 오퍼레이션의 경우에는 외부 시스템에 전달될 정보가 파라메터로 정의된다. ❑ '~결과수신' 오퍼레이션의 경우에는 파라메터가 없는 것이 일반적이다.

실체클래스 개요

- 실체(Entity)클래스는 영속적으로 유지되어야 하는 각 정보 항목에 대한 정의 뿐만 아니라, 이 정보에 대한 관리를 담당
  - 영속성이 있는 정보를 생성하고, 조회하고, 삭제하는 기능을 수행
- 대부분의 시스템에서 데이터베이스를 통해서 데이터의 영속성은 구현
- 실체클래스의 역할



- UI 경계클래스 : b1, b2, b3, b4
- SI 경계클래스 : b5
- 실체클래스 : e1, e2, e3

실체클래스 도출(1/2)

- 실체클래스는 문제기술서(Problem Statement)에서 명사를 분석하여 클래스를 도출하는 전통적인 방법을 통해서 도출
- 구축할 시스템에 대한 요구사항을 기술하고 있는 문제기술서에 등장하는 주요한 용어/개념들이 실체클래스에 대응될 가능성이 높음
- 문제기술서 부분1

> 수업담당자는 새로운 강좌를 등록할 수 있다. 강좌에 대해서는 강좌번호, 강좌이름, 담당학과, 학점 수 및 강좌에 대한 간단한 설명이 제공될 수 있어야 한다.

  - 문제기술서의 명사가 실체클래스 보다는 속성에 해당하는 경우
- 문제기술서 부분2

> 학적과의 학사관리담당자는 학생 및 교수 정보를 관리해야 한다. 즉, 새 학생/교수를 등록하거나 수정, 조회, 삭제를 지원해야 한다.

  - 문제기술서의 명사가 실체클래스 보다는 오퍼레이션에 해당하는 경우
    - 관리, 등록, 수정, 조회, 삭제
  - 문제기술서의 명사가 실체클래스 보다는 액터에 해당하는 경우
    - 학생, 교수, 수업담당자, 학사담당자, 직원
  - 실체클래스의 보다는 관련이름이 문제기술서에 명시적으로 언급되기 이 있는 속성들을 통합한 개념을 정의하는 방식으로 도출될 수 있음
    - 사용자정보라는 용어는 없지만, 아이디와 암호를 통칭하여 사용자정보라는 이름의 실체클래스를 정의

실체클래스 도출(2/2)

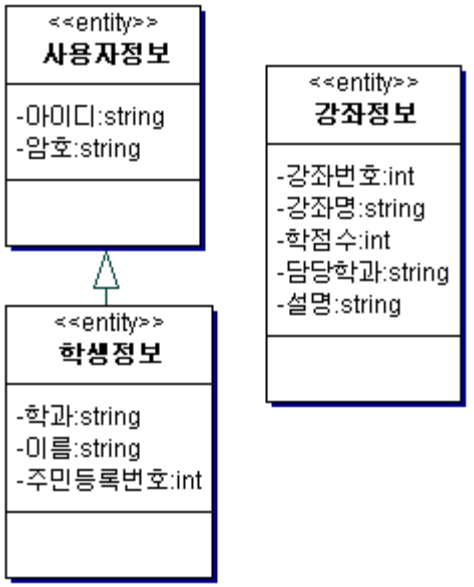
- **문제기술서 부분1**

> 학생/교수/직원은 등록된 아이디와 암호를 입력하여 시스템에 로그인 할 수 있다.

  - 명사 기반 클래스 도출 방법은 문제기술서 뿐만 아니라, 유스케이스명세서의 이벤트 흐름에도 적용하여 실체클래스를 파악하는 노력도 추가적으로 수행되어야 함
  - 실체클래스의 특징은 그 값이 대부분 사용자에 의해서 입력됨
  - 사용자에 의해서 입력되는 정보에 대응되는 실체클래스를 정의
- **실체클래스 간에는 연관, 포함, 일반화 등의 관계를 정의**
- **실체클래스의 예**

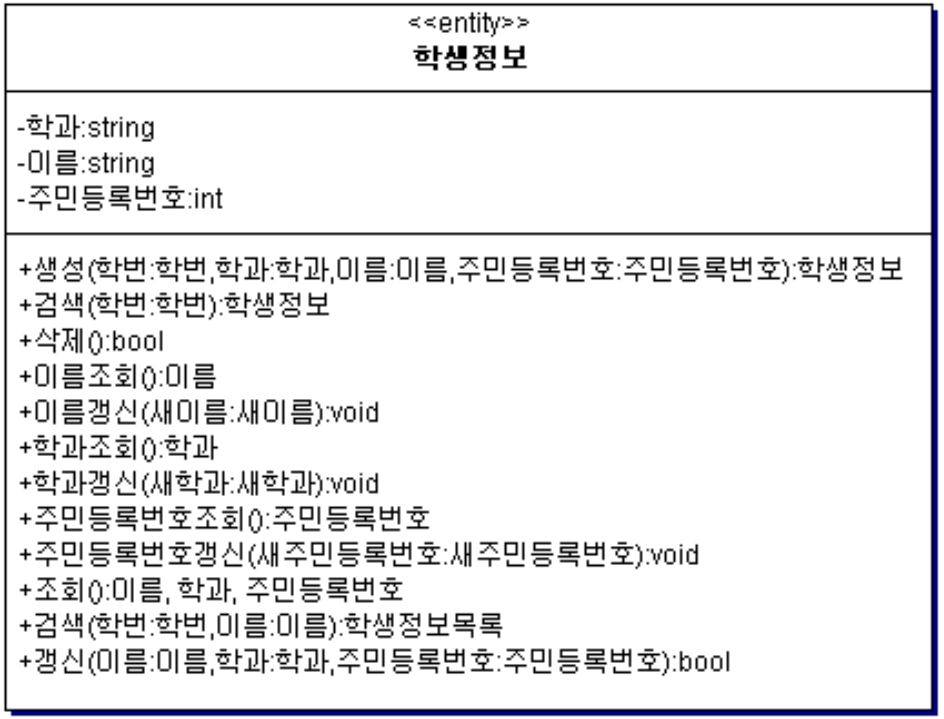
실체클래스 속성

- 속성은 실체클래스가 관리할 영속성이 있는 각 정보 항목
- 설계 활동에서 실체클래스는 관계형 DBMS를 이용하는 경우 테이블에 대응되며, 실체클래스의 속성은 테이블의 컬럼에 대응



실체클래스 오퍼레이션

- 실체클래스는 영속적인 정보에 대한 관리 기능을 담당
  - 영속적인 정보의 생성, 검색, 수정, 조회, 삭제 기능을 제공



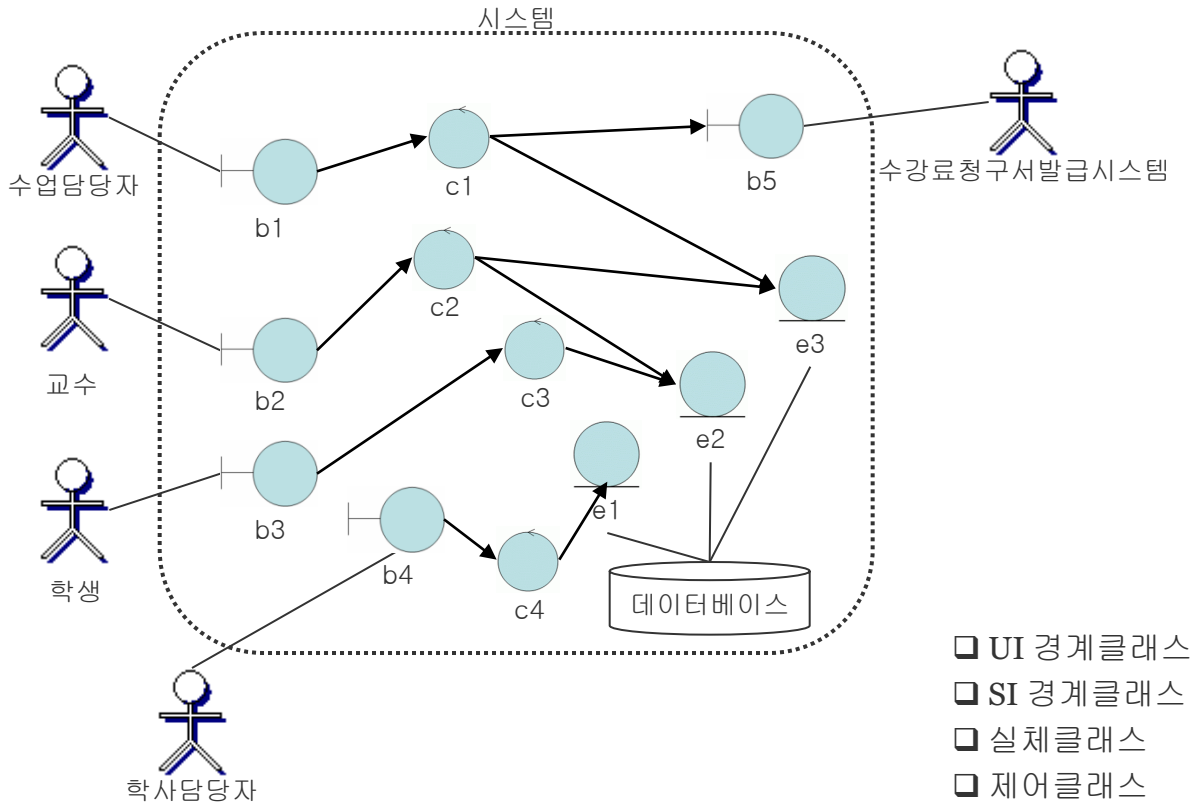
실체클래스 도출 요약

❑ 실체클래스의 도출방법

도출 방법	문제기술서와 유스케이스명세서를 바탕으로 분석하여 영속성이 필요한 정보를 추출한다.	
클래스 이름	'~정보'	
속성	영속성을 유지해야 하는 각 정보 항목을 속성으로 정의한다.	
오퍼레이션	객체의 생성, 검색, 삭제 및 각 속성에 대한 조회와 갱신을 지원한다.	
	생성	❑ 생성(속성1, 속성2, ..., 속성n): 실체클래스
	검색	❑ 검색(주요 키 속성): 실체클래스 ❑ 검색(속성1, 속성2, ...): 목록
	삭제	❑ 삭제(): Boolean
	조회	❑ "속성명" + 조회(): 속성명 ❑ 조회(): [속성1, 속성2, ..., 속성n]
	갱신	❑ "속성명" + 갱신(새 속성 값) ❑ 갱신(속성1, 속성2, ..., 속성n)

제어클래스 개요

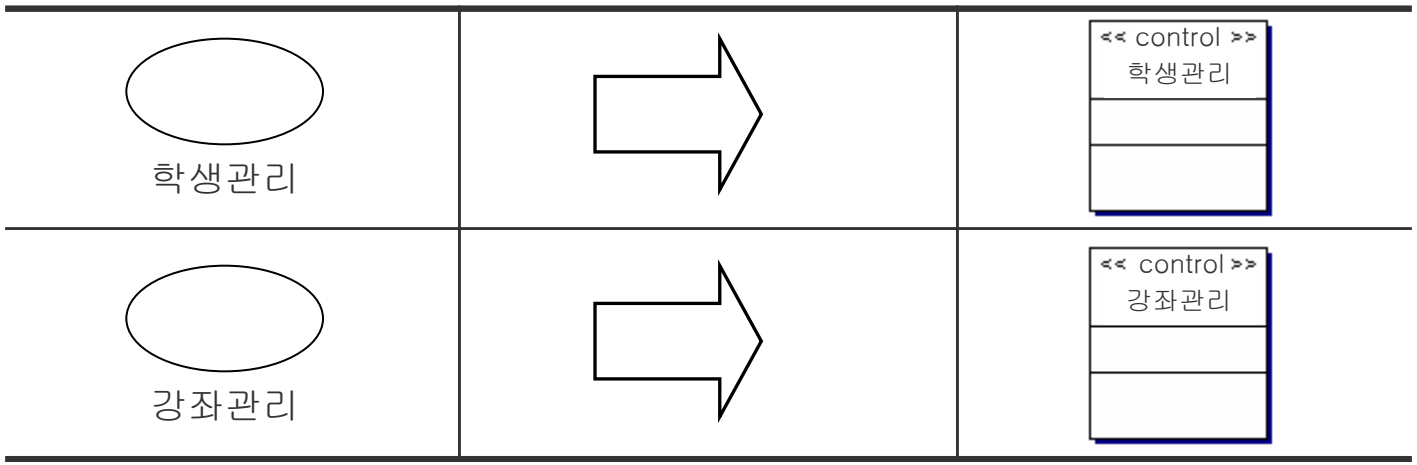
- 제어클래스는 시스템이 제공해야 하는 실질적인 기능, 비즈니스 로직을 제공하는 클래스
  - 사용자 인터페이스(UI 경계클래스), 외부시스템 인터페이스(SI 경계클래스), 데이터 접근 로직(실체클래스)을 제외한 모든 부분은 제어클래스의 책임



- UI 경계클래스 : b1, b2, b3, b4
- SI 경계클래스 : b5
- 실체클래스 : e1, e2, e3
- 제어클래스 : c1, c2, c3, c4

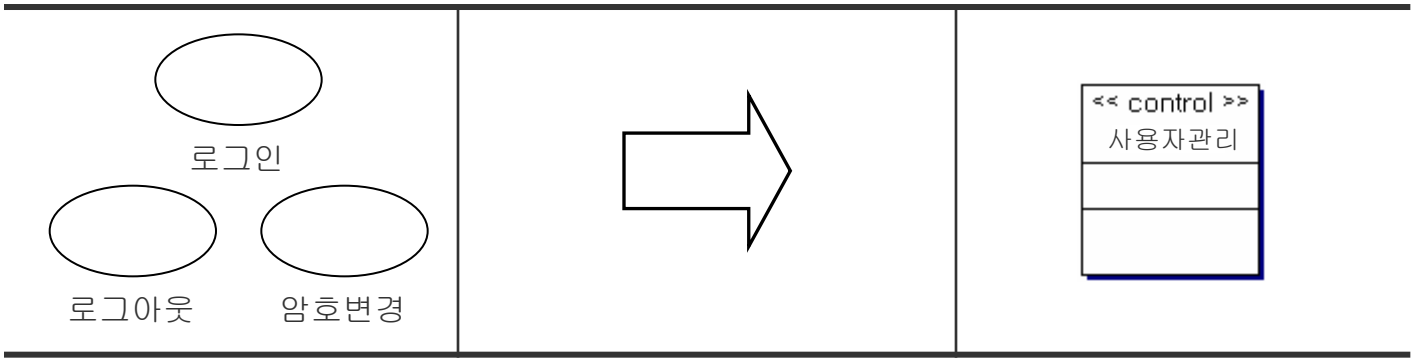
제어클래스 도출(1/2)

- 각 유스케이스별로 제어클래스로 정의
- 유스케이스는 시스템이 제공하는 독립적인 기능단위
- 유스케이스별로 제어클래스가 정의되므로 유스케이스 이름을 그대로 제어클래스의 이름으로 주는 것이 바람직함



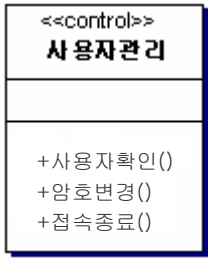
제어클래스 도출(2/2)

- 유스케이스는 비교적 작지 않은 기능 단위이므로 유스케이스 당 하나의 제어클래스가 대부분의 경우에 적절함
  - 관련성이 높은 작은 규모의 여러 유스케이스에 대해서는 하나의 제어클래스를 정의하고 이를 두 유스케이스들이 공유하는 것이 바람직함
    - 로그인 유스케이스, 암호변경 유스케이스, 로그아웃 유스케이스
- 각 유스케이스 별로 제어클래스를 정의하는 대신에 세 유스케이스의 기능을 모두 포함하는 사용자관리 클래스를 정의하고, 세 유스케이스가 이 하나의 사용자관리 제어클래스를 이용 하는 것이 바람직함



제어클래스 오퍼레이션(1/2)

- 제어클래스가 제공하는 유스케이스 기능
- UI경계클래스는 관련된 제어클래스의 오퍼레이션을 호출하면서 사용자가 입력한 값을 파라메터로 전달
- 다른 유스케이스를 분석하는 과정에서 각 유스케이스에 대응되는 새로운 제어클래스가 도출되기도 하지만, 기존의 제어클래스에 오퍼레이션이 추가될 수도 있음



사용자관리 제어클래스의 오퍼레이션 - 초안

사용자관리 제어클래스의 오퍼레이션 - 구체화

사용자관리 제어클래스의 오퍼레이션 - 완성

제어클래스 오퍼레이션(2/2)

- 제어클래스의 오퍼레이션에서 해당 유스케이스의 기능을 실제로 제공할 때 영속적인 데이터를 조작하는 부분은 실체클래스의 오퍼레이션을 호출
- 사용자관리 클래스와 사용자정보 클래스의 오퍼레이션

사용자관리 제어클래스	사용자정보 실체클래스
등록()	생성()
사용자확인()	암호조회()
암호변경()	암호갱신()
삭제()	삭제()

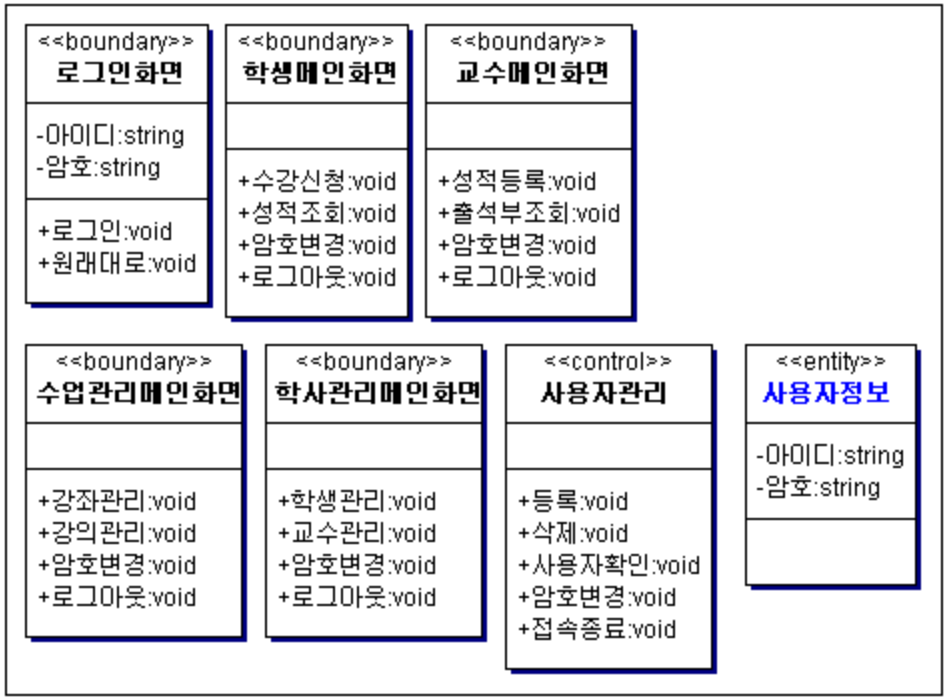
제어클래스 도출 요약

❑ 제어클래스의 도출방법

도출 단위	❑ 각 유스케이스 마다 별도의 제어클래스 정의 ❑ 단, 작으면서 관련성이 높은 유스케이스들은 하나의 제어클래스를 공유할 수 있다.	
클래스이름	유스케이스와 동일한 이름을 사용한다.	
속성	필요한 모든 데이터는 경계클래스로부터 파라메터를 받고, 제어클래스의 오퍼레이션들 사이에는 공유할 데이터가 없으므로 속성은 정의되지 않는다.	
오퍼레이션	유스케이스가 나타내는 기능에 대응되며, 경계클래스로부터 호출된다.	
	이름	데이터 관점이 아니라, 사용자 즉 경계클래스의 관점에서 명명해야 한다.
	접근권한	공용
	반환 타입	Boolean 등 오퍼레이션 별로 적절한 타입 사용
	파라메터	사용자가 화면을 통해서 입력한 값이 파라메터로 들어온다.

로그인 유스케이스의 분석클래스 요약(1/2)

❑ 로그인 유스케이스의 분석 클래스



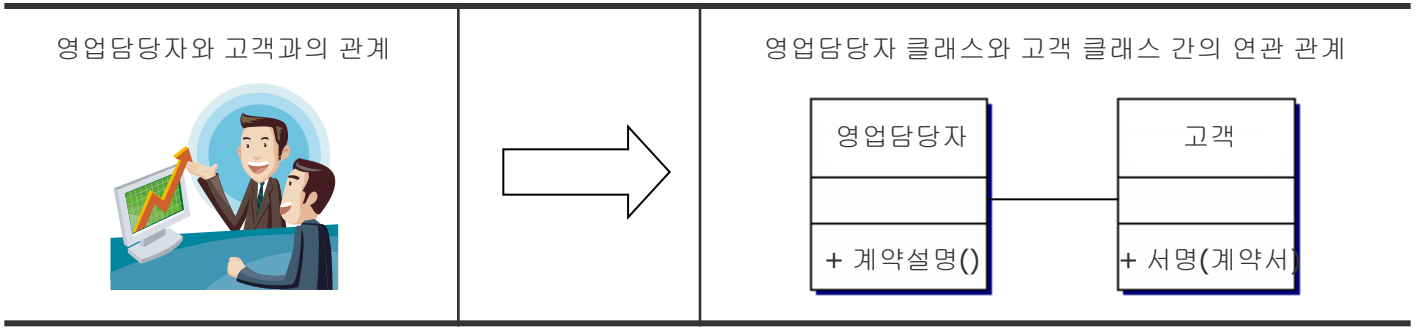
로그인 유스케이스의 분석클래스 요약(2/2)

- **하나의 유스케이스가 나타내는 시스템의 기능중에서 외부 시스템과의 연결 로직은 경계클래스, 비즈니스 로직은 제어클래스, 데이터 접근 로직은 실체클래스가 담당**
  - 하나의 유스케이스에 대하여 세 가지 유형의 분석 클래스가 도출
- **유스케이스 별로 사용자 인터페이스를 위하여 사용되는 각 화면마다 UI 경계클래스를 정의**
  - 개발하는 시스템과 연결되는 외부 시스템 마다 하나의 SI 경계클래스를 정의
  - 유스케이스 마다 별도의 제어클래스를 정의하는 것이 일반적
  - 실체클래스는 영속성이 필요한 정보 마다 정의되며, 논리적인 수준의 개체(entity) 또는 테이블과 유사
- **일반적으로 유스케이스를 분석할 때 마다 새로운 UI 경계클래스와 제어클래스가 정의**
  - 각 유스케이스별로 사용되는 화면과 비즈니스 로직이 다르기 때문
  - 하나의 영속성이 있는 데이터가 여러 유스케이스에서 이용되거나 조작될 수 있으므로, 실체클래스는 여러 유스케이스에서 공통적으로 이용될 수 있음

연관 관계(1/5)

❑ 연관관계의 의미

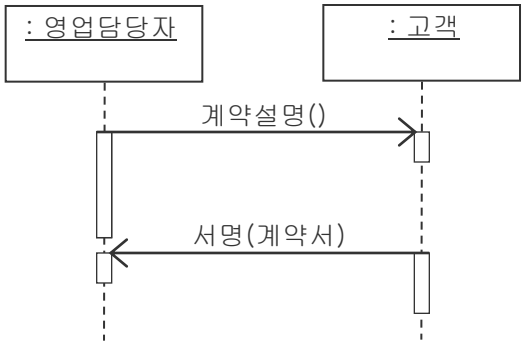
- 연관(Association)관계는 클래스 사이의 가장 일반적인 관계로 한 클래스가 다른 클래스를 인지(acquaintance)함을 의미
- 영업담당자 클래스의 계약설명() 오퍼레이션은 영업담당자가 고객에게 계약 사항에 대한 설명을 하는 행위를 나타냄
- 고객 클래스의 서명(계약서) 오퍼레이션은 고객이 계약서에 서명하는 행위



연관 관계(2/5)

❑ 연관관계의 용도

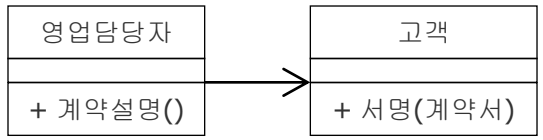
- 연관 관계는 상대클래스의 객체에게 메시지를 전달할 때 사용되는 통로 역할
  - 연관 관계를 맺고 있는 클래스 사이에 메시지가 전송할 수 있음



연관 관계(3/5)

❑ 연관관계의 구현 예

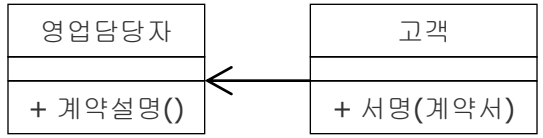
(1)




```
class 영업담당자 {
        private 고객 the_고객;
        public 계약설명();
}
```

```
class 고객 {
        public 서명(계약서);
}
```

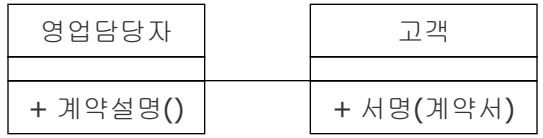
(2)




```
class 영업담당자 {
        public 계약설명();
}
```

```
class 고객 {
        private 영업담당자 the_영업담당자;
        public 서명(계약서);
}
```

(3)

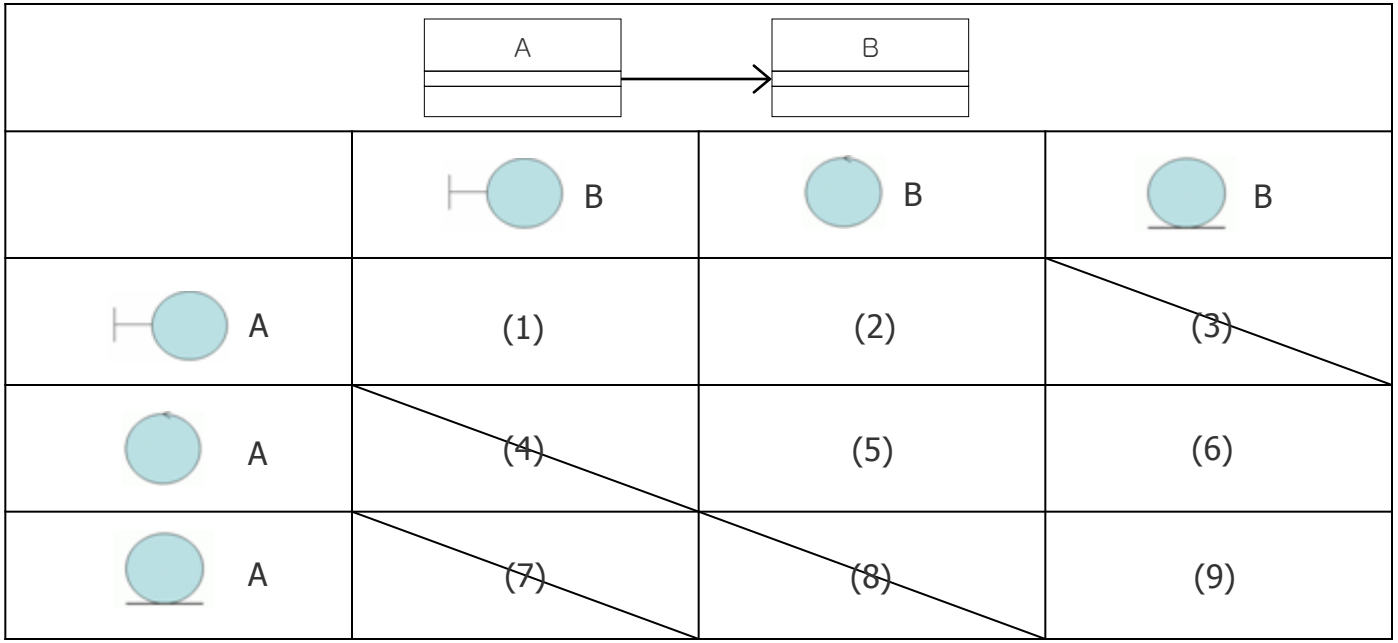



```
class 영업담당자 {
        private 고객 the_고객;
        public 계약설명();
}
```

```
class 고객 {
        private 영업담당자 the_영업담당자;
        public 서명(계약서);
}
```

연관 관계(4/5)

- 분서클래스간의 연간 관계는 일정한 형태로만 발생한다.
  - 제어클래스에서 실체클래스로의 연관은 가능하지만 반대로 실체클래스에서 제어클래스로의 연관은 허용되지 않음
- 분석클래스 간의 연관관계



연관 관계(5/5)

❑ 포함 관계에 있는 유스케이스