

V. 유스케이스

유스케이스 기반의 요구사항 분석

의미

- 요구사항 명세서 작성의 기반을 다지는 작업

요구사항 분석 방법

- 객체지향 방법인 유스케이스 기반 분석

유스케이스 모델링



기술서 작성



기능,비기능 분류



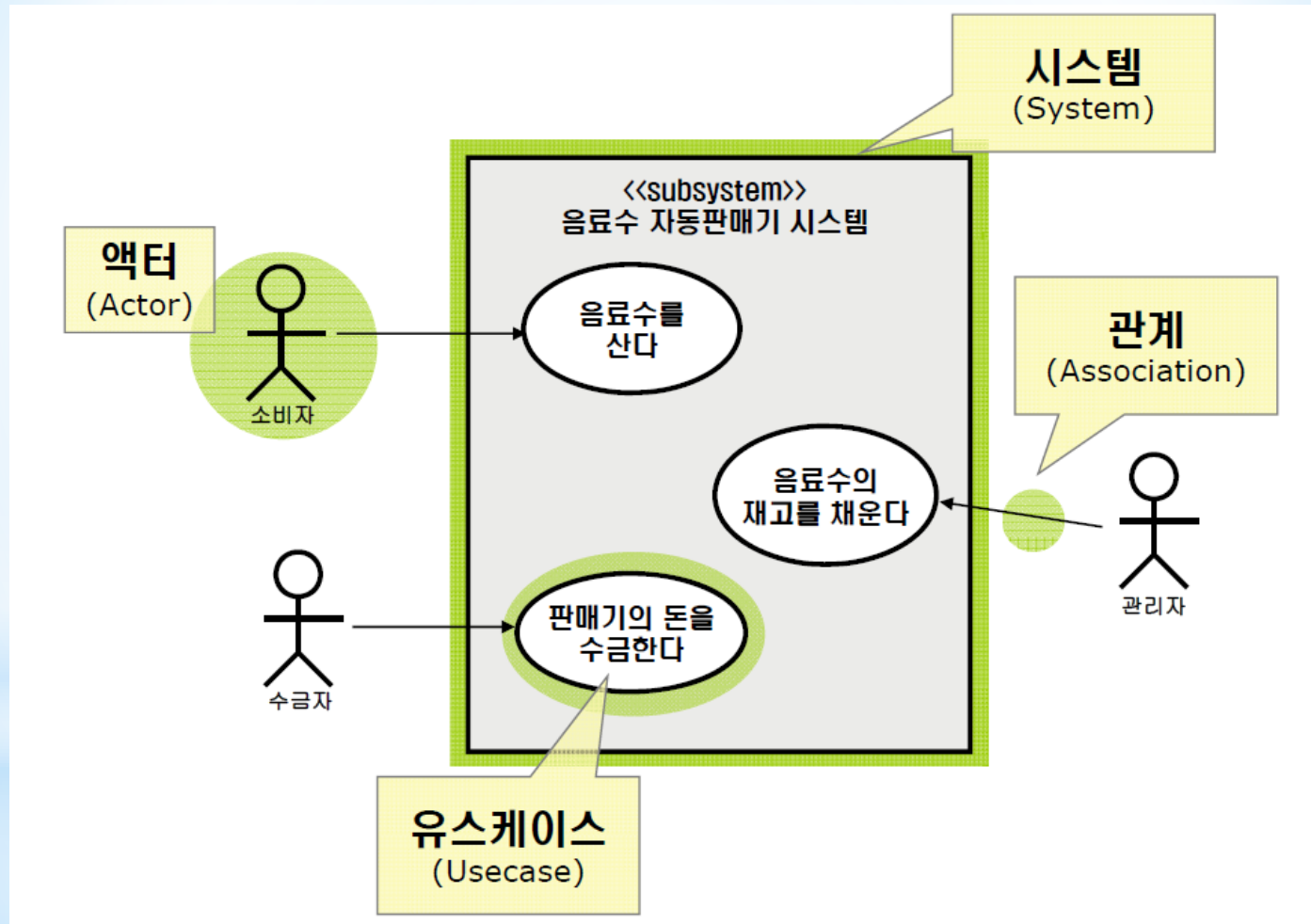
요구사항명세서 작성

유스케이스 다이어그램

개요

- 사용자의 관점에서 시스템의 서비스 혹은 기능 및 그와 관련한 외부 요소를 보여주는 다이어그램
- 고객과 개발자가 함께 보며 요구사항에 대한 의견을 조율할 수 있음

유스케이스 다이어그램의 구성요소



유스케이스 다이어그램의 구성요소

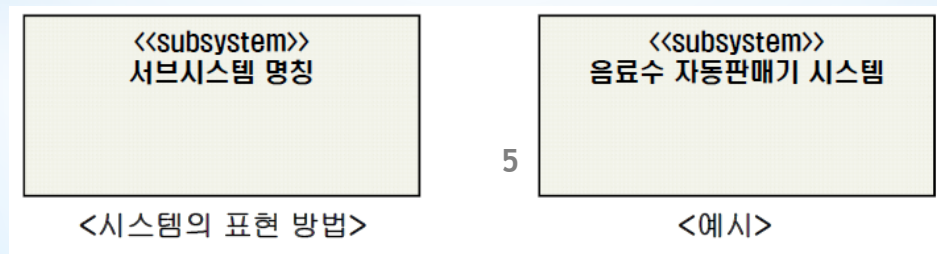
시스템 (System)

의미

- 만들고자 하는 시스템의 범위

표기법

- 유스케이스나 액터를 둘러싼 사각형의 틀을 그리고, 시스템이나 모델의 명칭을 사각형 안쪽 상단에 기술
- 서브시스템일 경우<<subsystem>>이라 기술하고 모델(액터, 유스케이스)의 단위일 경우에 <<useCaseModel>>이라고 기술한다



유스케이스 다이어그램의 구성요소

액터 (Actor)

의미

- 시스템의 외부에 있으면서 시스템과 상호 작용을 하는 사람 또는 다른 시스템

표기법

- 원과 선을 조합하여 사람 모양으로 표현
- 그 위 또는 아래에 액터명 표시
- 액터명은 액터의 역할로 정함



유스케이스 다이어그램의 구성요소

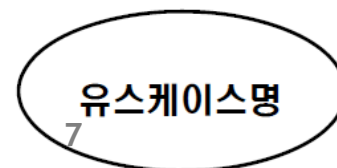
유스케이스(Usecase)

의미

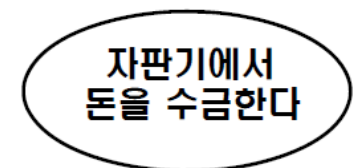
- 시스템이 액터에게 제공해야 하는 기능의 집합
- 시스템의 요구사항을 보여줌

표기법

- 타원으로 표시하고 그 안쪽이나 아래쪽에 유스케이스 이름을 기술
- 유스케이스의 이름은 “~한다” 와 같이 동사로 표현
- 각 유스케이스가 개발될 기능 하나와 연결될 수 있도록 한다.



<유스케이스의 표현 방법>



<예시>

유스케이스 다이어그램의 구성요소

관계 (Relationship)

의미

- 액터와 유스케이스 사이의 의미 있는 관계

종류

- 연관 관계 (Association)
- 의존 관계
 - 포함 관계 (include)
 - 확장 관계 (extend)
- 일반화 관계 (generalization)

유스케이스 다이어그램의 구성요소

관계 (Relationship)

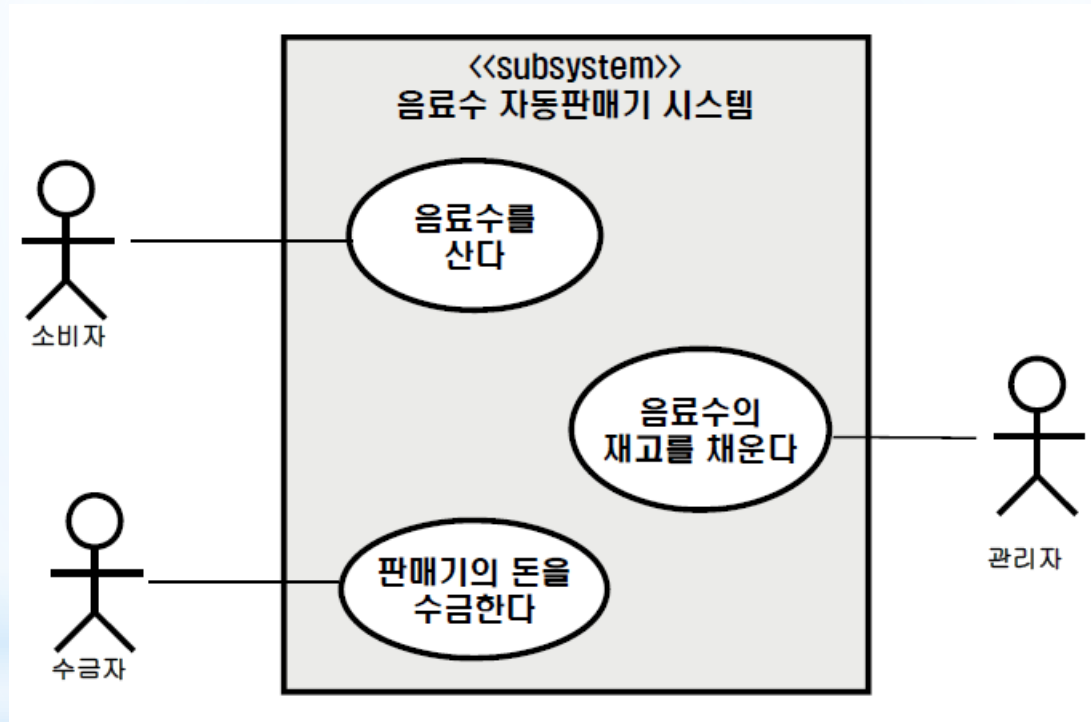
관계 종류	설명	표기법
연관 관계	<ul style="list-style-type: none"> • 유스케이스와 액터간의 상호작용이 있음을 표현 • 유스케이스와 액터를 실선으로 연결함 	
포함 관계 (include)	<ul style="list-style-type: none"> • 유스케이스가 다른 유스케이스를 포함하는 경우 • 포함되는 유스케이스는 포함하는 유스케이스를 실행하기 위해 반드시 실행되어야 하는 유스케이스 • 포함하는 쪽에서 포함되는 쪽으로 점선으로 된 화살표를 그리고, <<include>>라는 표시 한다. 	

유스케이스 다이어그램의 구성요소

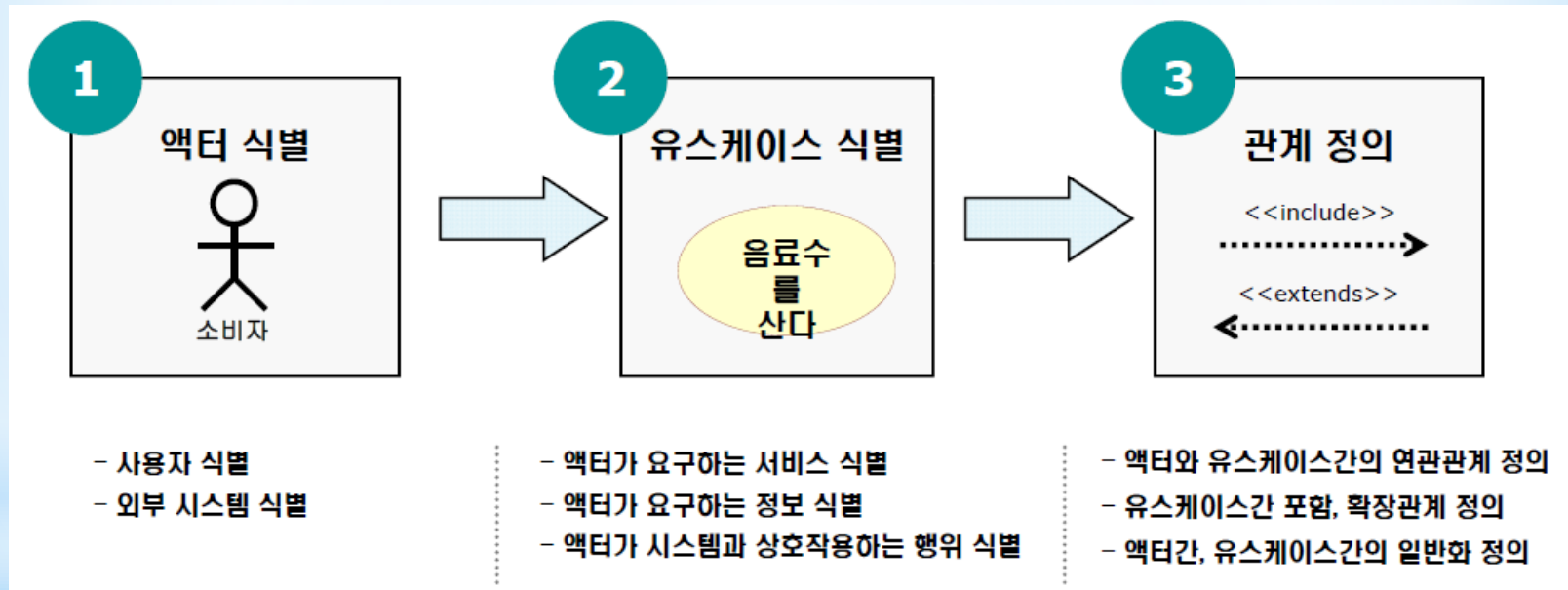
관계 (Relationship)

관계 종류	설명	표기법
확장 관계 (extend)	<ul style="list-style-type: none"> 어떠한 유스케이스로부터 다른 유스케이스가 특정 조건에서 생성되는 경우 확장 기능 유스케이스는 특정 조건이나 액터의 선택에 따라 발생 확장된 기능을 가지는 유스케이스에서 확장 대상이 되는 원래의 기능을 가진 유스케이스 쪽으로 점선 화살표를 그리고, <<extend>>라고 함께 표시한다. 	
일반화 관계 (generalization)	<ul style="list-style-type: none"> 액터나 유스케이스가 구체화 된 다른 여러 액터나 유스케이스로 구성될 경우 구체적인 유스케이스로부터 추상적인 유스케이스 쪽으로 속이 비어있는 삼각형 모양으로 된 실선 화살표를 그려 표현한다. 	

단순화된 유스케이스 다이어그램의 예



유스케이스 다이어그램 작성 순서



액터 식별

액터를 찾기 위한 질문들

- 누가 정보를 제공하고, 사용하고, 삭제하는가?
- 누가 또는 어떤 조직에서 개발될 시스템을 사용할 것인가?
- 누가 요구사항에 대해 관심을 가지고, 시스템이 만들어낸 결과에 관심이 있는가?
- 누가 시스템이 잘 운영될 수 있도록 유지보수 및 관리를 하는가?
- 개발될 시스템과 상호작용하는 하드웨어나 소프트웨어 시스템은 무엇인가?

유스케이스 식별

유스케이스를 찾기 위한 질문들

- 액터가 원하는 시스템 제공 기능은 무엇인가?
- 액터는 시스템에 어떤 정보를 생성, 수정, 조회, 삭제 하고 싶어 하는가?
- 액터는 시스템의 갑작스러운 외부 변화에 대해 어떤 정보를 필요로 하는가?
- 시스템이 어떤 기능을 제공하면 액터의 일상 작업이 효율적이고 편리해지는가?
- 모든 기능 요구사항들을 만족할 수 있도록 유스케이스가 모두 식별되었는가?

관계 식별

연관 관계 (Association)

- 액터와 유스케이스 간에 상호 작용이 존재하는가?

포함 관계 (Include)

- 이 유스케이스를 실행하기 위하여 반드시 실행되어야 하는 유스케이스가 존재하는가?

확장관계 (Extend)

- 이 유스케이스를 실행하기 위하여 기존 유스케이스를 참조하는가?

일반화 관계 (generalization)

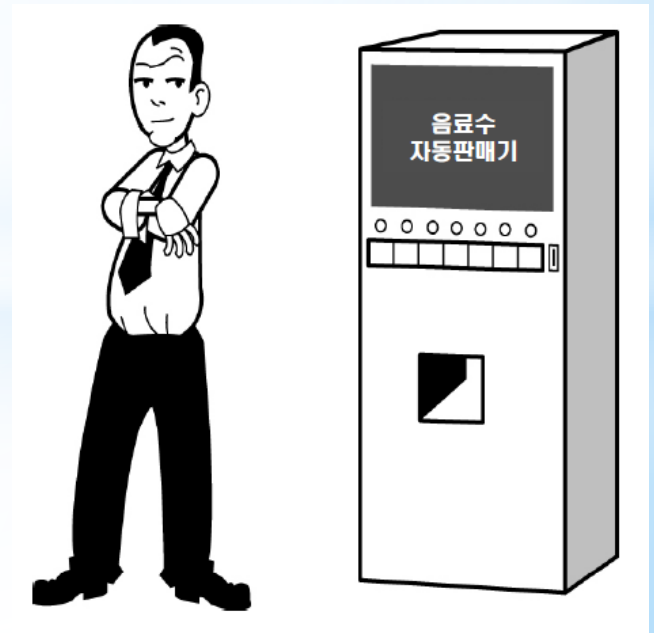
- 액터 또는 유스케이스가 구체화 된 다른 여러 액터나 유스케이스를 가지고 있는가?

유스케이스 다이어그램의 예

예제 요구사항

- SE사는 K고객으로부터 다음의 요구사항을 전달 받았다.
- 음료수 자동판매기 시스템을 만드시오.

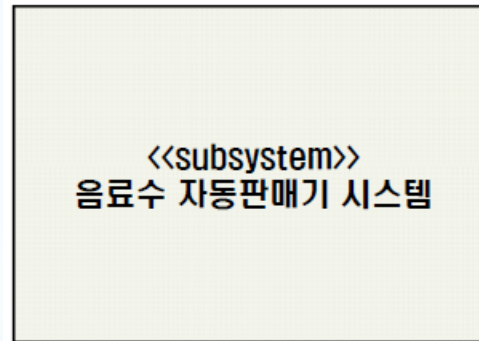
SE사는 K고객의 요구사항을 Usecase Diagram으로 모델링하기로 한다.



유스케이스 다이어그램의 예

시스템 식별

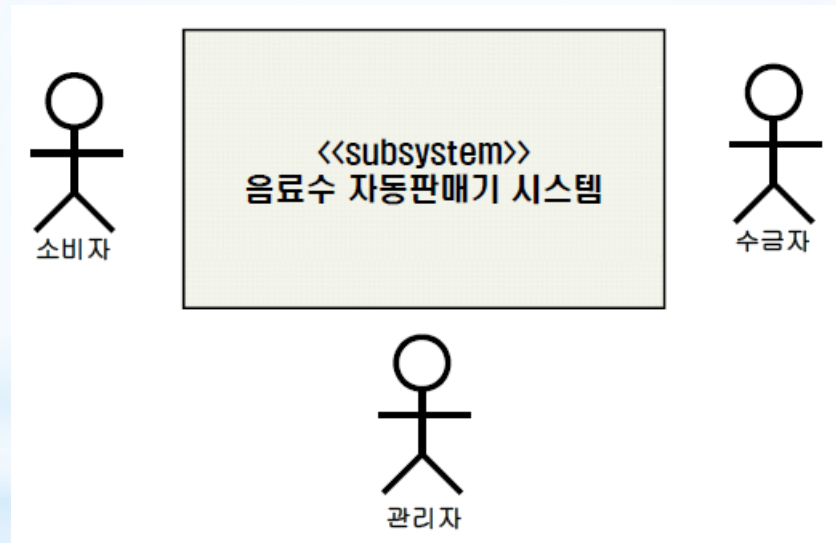
- 요구사항을 통해 만들고자 하는 시스템은
“음료수 자동판매기 시스템”



유스케이스 다이어그램의 예

액터 식별

- 음료수 자동판매기(시스템) 외부에서 상호작용하는 액터는 소비자, 관리자, 수금원으로 식별할 수 있다.



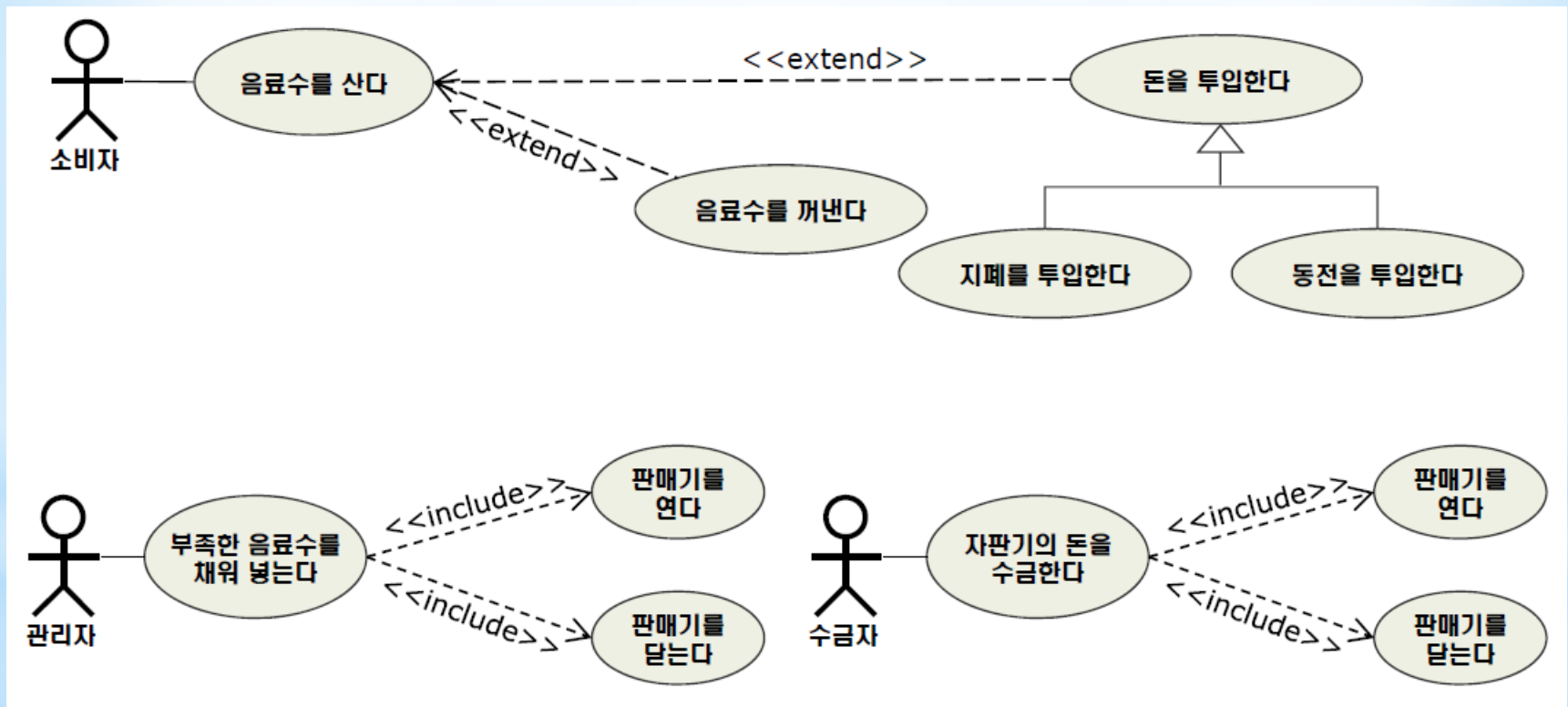
유스케이스 다이어그램의 예

유스케이스 식별



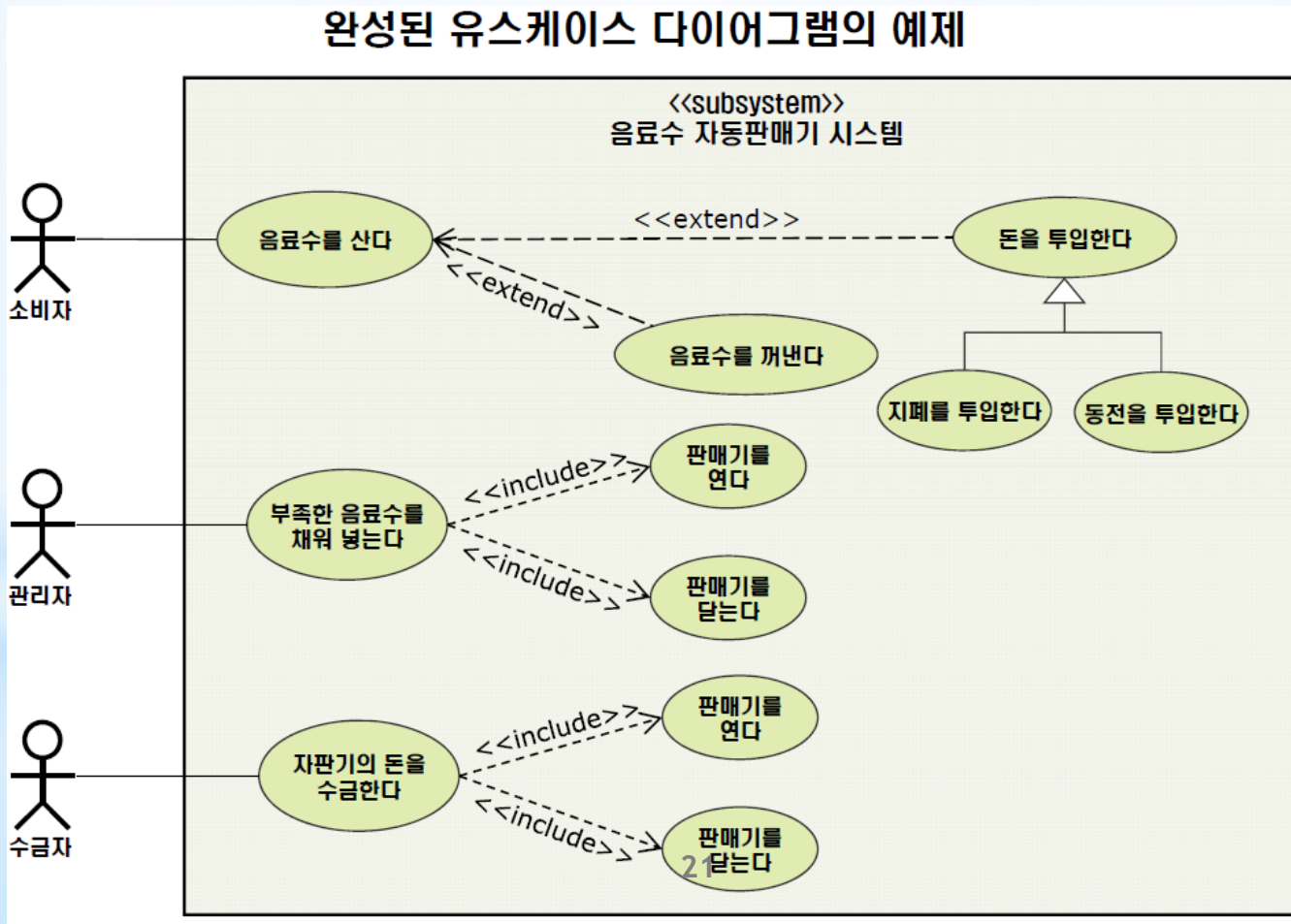
유스케이스 다이어그램의 예

관계 정의



유스케이스 다이어그램의 예

완성된 유스케이스 다이어그램의 예제



유스케이스 기술서 작성

개요

- 유스케이스 다이어그램을 보완하기 위한 산출물
- 유스케이스 다이어그램과의 차이
 - 유스케이스 다이어그램: 유스케이스는 시스템의 기능을 표현하는 것
 - 유스케이스 기술서: 각각의 유스케이스에 대해서 해당 유스케이스가 어떻게 수행되는지를 표현하는 수단

유스케이스 기술서 작성

유스케이스 기술서 항목

- ❖ 유스케이스 명
- ❖ 액터 명
- ❖ 유스케이스 개요 및 설명
- ❖ 사전 및 사후 조건
- ❖ 작업 흐름
 - 정상흐름(Normal Flow): 해당 유스케이스가 정상적으로 수행되는 흐름을 표현하는 절차
 - 대치 흐름(Alternative Flow): 유스케이스 내의 작업 흐름이 수행되는 중에 특정 시점에서 여러 가지 선택적인 흐름으로 나뉘어질 경우에 발행하는 흐름
 - 예외 흐름(Exceptional Flow): 유스케이스 내의 작업 흐름이 수행되는 중에 발생할 수 있는 예외 상황이나 오류를 표현하는 흐름
- ❖ 시나리오: 각 시나리오는 유스케이스의 특정한 예를 나타냄

유스케이스 기술서 작성

유스케이스 기술서 예제

- ❖ **유스케이스명: 음료수 보충**
- ❖ **액터명: 관리자**
- ❖ **유스케이스 개요 및 설명**
 - 음료수 관리자는 2주마다 음료수 자동판매기의 부족한 음료수를 보충한다.
- ❖ **사전 조건: 마지막으로 음료수를 보충한지 2주가 지났다.**
- ❖ **작업 흐름**
 - **정상흐름**
 - 1. 유스케이스는 2주마다 시작한다.
 - 2. 관리자는 음료수 자동판매기를 연다.
 - 3. 부족한 음료수를 보충한다.
 - 5. 관리자는 음료수 자동판매기를 닫고 유스케이스는 종료된다.
- ❖ **사후 조건: 음료수 자동판매기에 부족한 음료수가 보충되었다.**

유스케이스 다이어그램의 예

인터넷쇼핑몰 유스케이스 모델링

인터넷쇼핑몰 문제설명서 (Internet Shopping Mall Problem Statement)

인터넷쇼핑몰 시스템은 상품의 판매나 구입을 간편하게 하기 위한 시스템이다. 쇼핑몰 관리자는 인터넷쇼핑몰에 상품을 등록해서 상품의 판매를 쉽게 진행할 수 있으며, 구매자 또한 인터넷쇼핑몰에 등록된 상품들 중에서 구매하고자 하는 상품을 간편하게 구매할 수 있다.

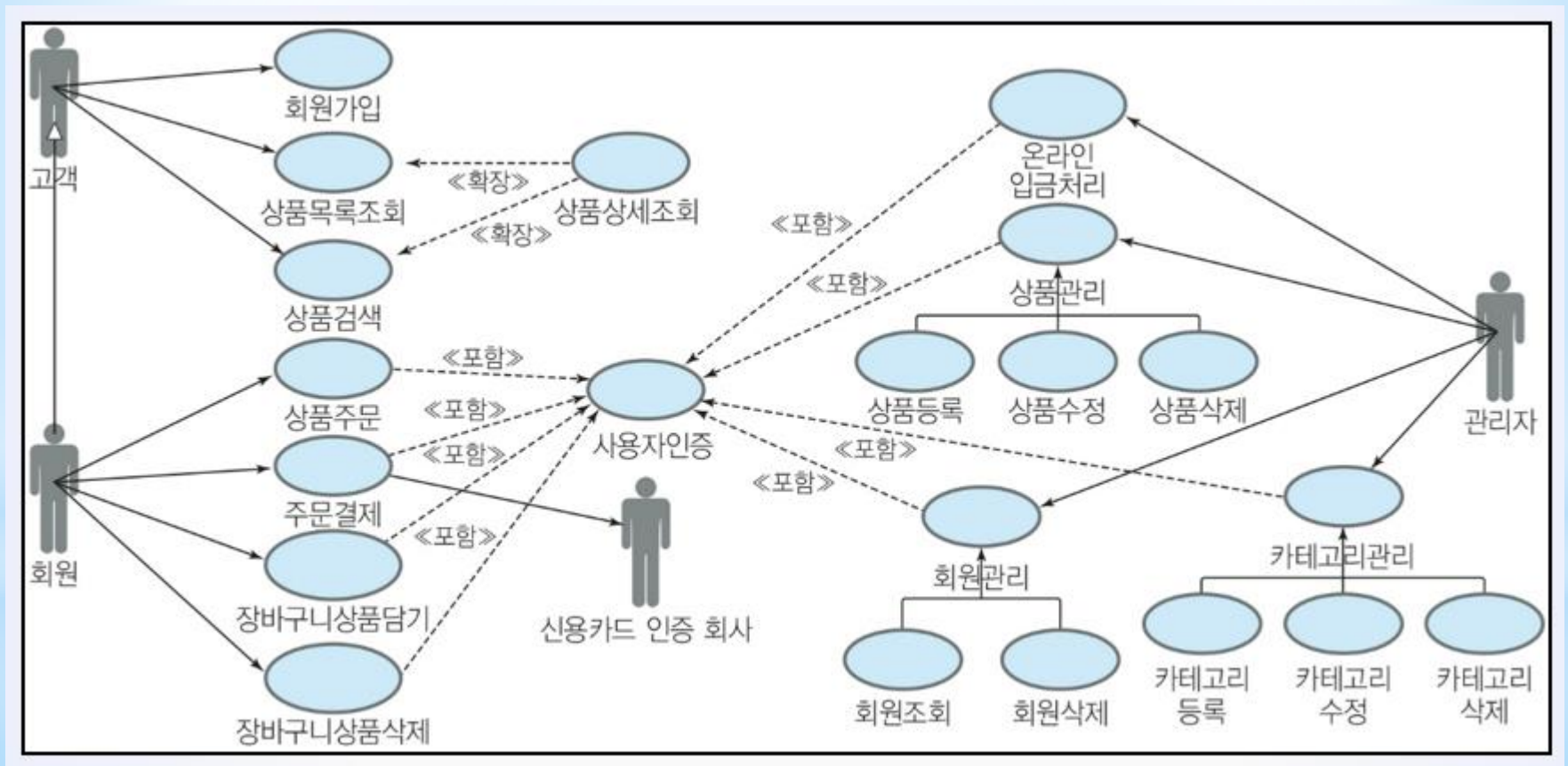
인터넷쇼핑몰을 사용하는 사용자는 일반 고객(비회원), 회원, 관리자로 나뉜다. 일반 고객은 상품 구매와 같은 기능을 수행하기 위해서 회원으로 가입을 할 수 있다. 관리자는 상품 관리(등록, 수정, 삭제)나 회원 관리(조회, 삭제), 카테고리 관리 등과 같은 활동을 수행한다.

상품에 대한 조회 및 검색은 쇼핑몰에 가입한 회원뿐만 아니라 가입하지 않은 비회원 및 관리자도 수행할 수 있다. 상품 조회는 특정 카테고리의 선택 과정을 통해 이루어지며, 상품검색은 특정 검색어의 입력을 통해 이루어진다. 관리자는 카테고리의 분류 체계를 관리하고, 상품을 특정 카테고리에 포함시켜 등록한다.

상품구매는 회원만 할 수 있고, 상품을 구매하기 위해서는 먼저 원하는 상품을 장바구니에 담고, 장바구니에 담긴 상품들 중에서 선택적으로 주문할 수 있다. 상품 구매 시 결제 수단은 온라인 입금과 신용카드 결제로 구분한다. 온라인 입금은 뱅킹 시스템을 통해 입금 여부를 확인한 후 결제 처리하며, 신용카드 결제는 신용카드 인증회사로의 결제승인 서비스를 통해 즉시 처리가 가능하도록 한다.

유스케이스 다이어그램의 예

인터넷 쇼핑몰 유스케이스 다이어그램



V. UML 클래스 다이어그램

UML(Unified Modeling Language)

Q 분석, 설계를 비주얼 화, 문서화 하기 위한 그래픽 언어

Q **Unified**

- 이전의 OO 방법들의 통합

Q **Modeling**

- 객체지향 분석 설계를 위한 비주얼 모델링

Q **Language**

- 모형화된 지식(의미)을 표현



UML(Unified Modeling Language)

UML은 () 이다.

- Q 시스템에 대한 지식을 찾고 표현하기 위한 언어
- Q 시스템을 개발하기 위한 탐구 도구
- Q 비주얼 모델링 도구
- Q 근거가 잘 정리된 가이드라인
- Q 분석, 설계 작업의 마일스톤
- Q 실용적 표준

UML(Unified Modeling Language)

UML은 () 이 아니다.

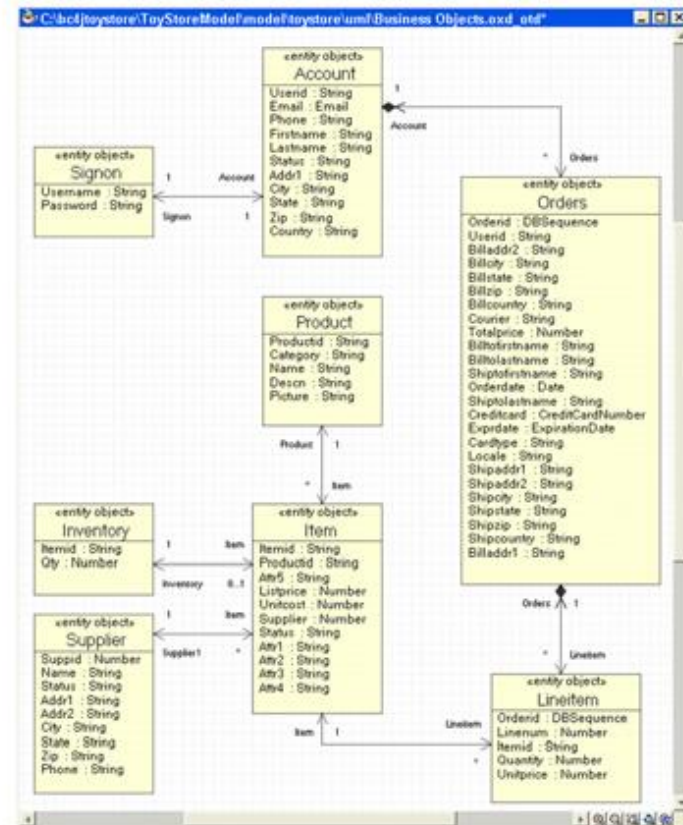
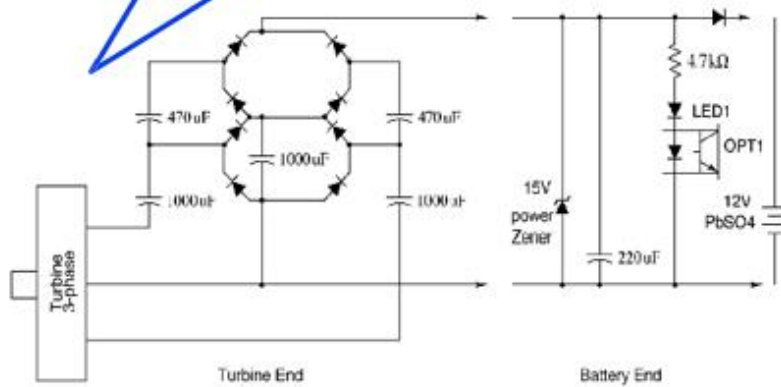
- Q 비주얼 프로그래밍 언어 환경
- Q 데이터베이스 표현 도구
- Q 개발 프로세스(SDLC)
- Q 모든 문제의 해결책
- Q 품질 보증 방안

UML(Unified Modeling Language)

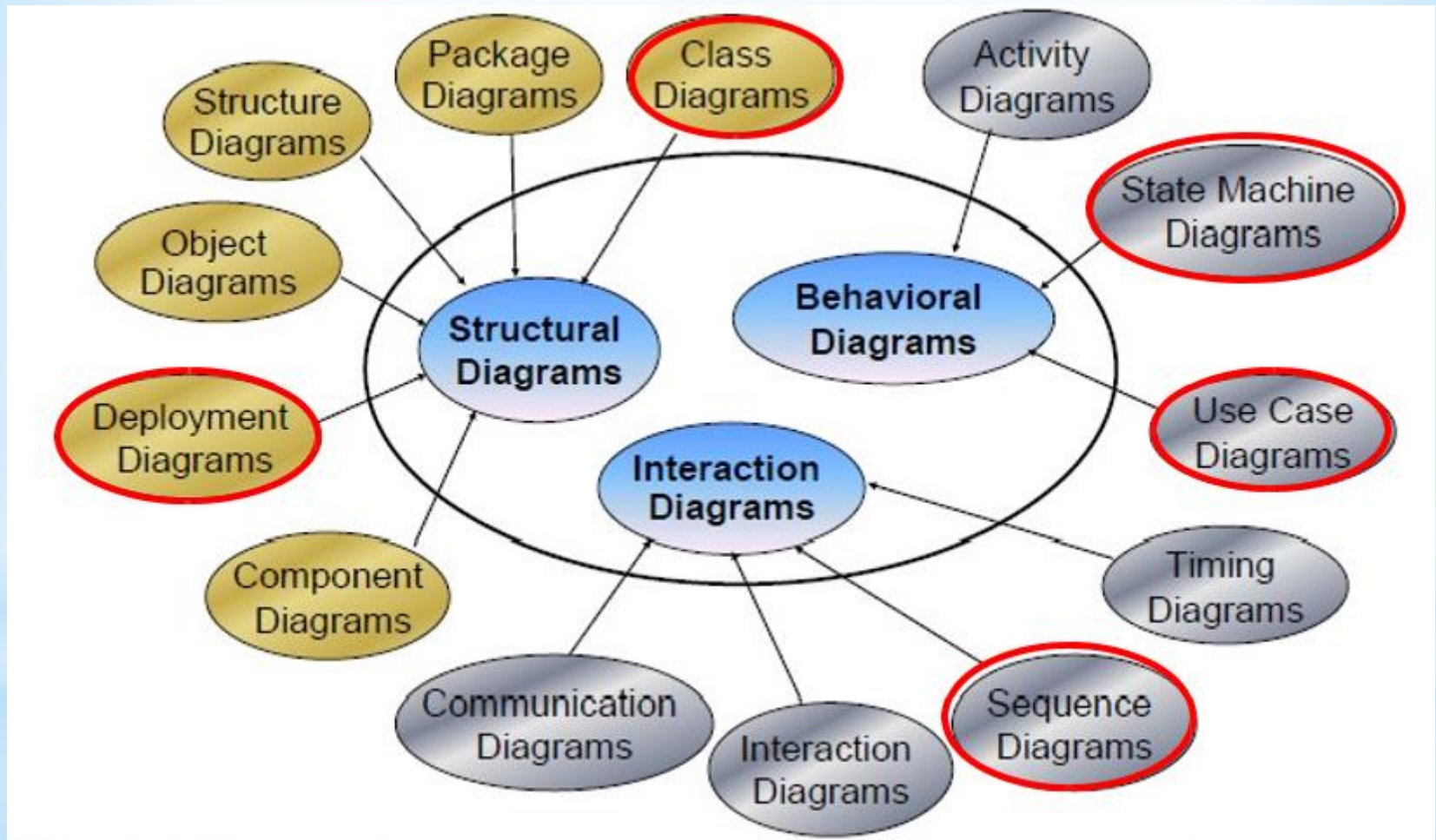
Every s/w engineer
WILL understand UML
diagrams.



Every h/w engineer
understands circuit
diagram.



UML 2.0 다이어그램 체계



UML 클래스 다이어그램

Q UML 클래스 다이어그램

객체지향 시스템에 존재하는 클래스, 클래스 안의 필드, 메소드, 서로 협력하거나 상속하는 클래스 사이의 연결 관계를 나타내는 그림

나타내지 않는 것

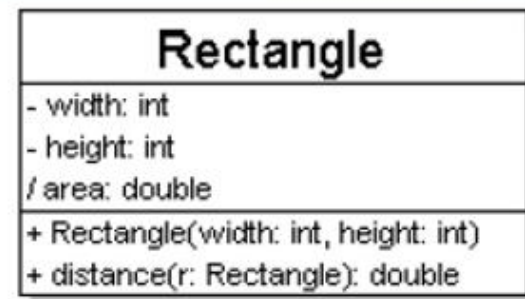
- 클래스가 서로 어떻게 상호작용 하는지
- 자세한 알고리즘
- 특정한 동작이 어떻게 구현되는지

UML 클래스 다이어그램

클래스 나타내기

Q 박스 위에 클래스 이름

- 추상 클래스는 이탤릭체
- 인터페이스 클래스는 <<interface>>

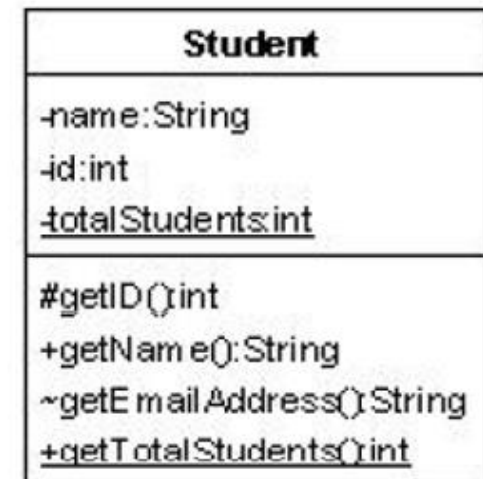


Q 속성

- 객체가 가지는 모든 필드를 포함

Q 오퍼레이션/메소드

- 아주 흔한 메소드(get/set)는 생략
- 상속된 메소드도 포함할 필요 없음



UML 클래스 다이어그램

클래스 속성

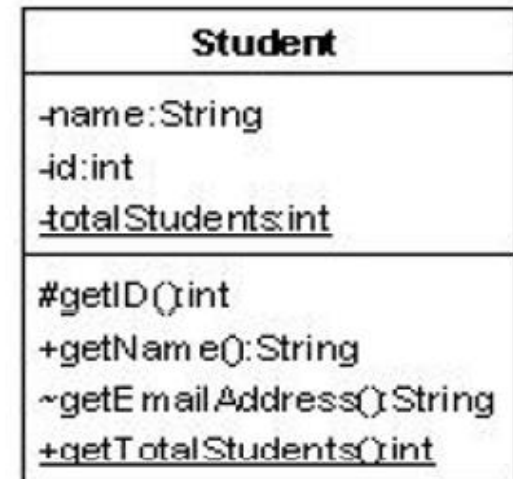
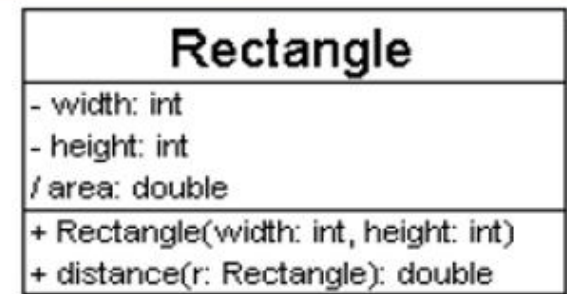
Q 속성(필드, 인스턴스 변수)

○ **visibility** name: **type**[count] = *default value*

- visibility: + public
- # protected
- private
- ~ package(디폴트)
- / derived

○ Underline static variable

○ 파생된 속성: 저장되지 않고 다른 속성값으로부터 계산됨



UML 클래스 다이어그램

클래스 오퍼레이션/메소드

Q 오퍼레이션/메소드

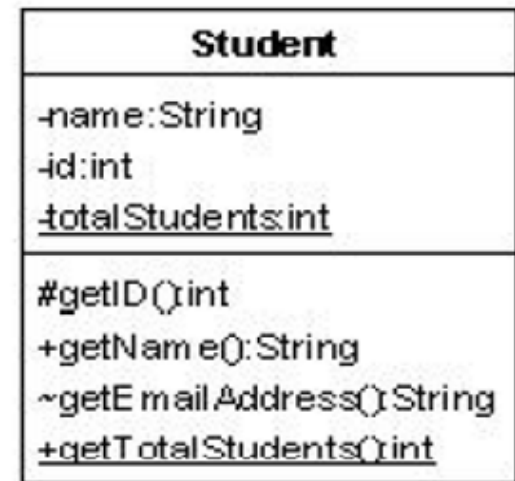
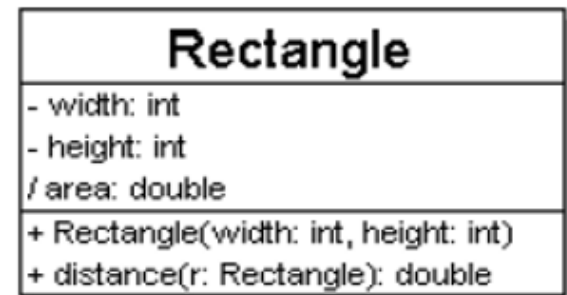
○ **visibility** **name(parameters)** : *return_type*

○ visibility: + public protecte
 # d private pack
 - age(디폴트)
 ~

○ Underline static method

○ 파라메타 타입 (name: type)

○ 생성자나 리턴 타입이 void인 경우는
return_type 생략



UML 클래스 다이어그램

클래스 사이의 관계

Q 일반화(generalization): 상속(isa) 관계

- 클래스 사이의 상속
- 인터페이스 구현

Q 연관(association): 사용(usage) 관계(3 종류)

- 의존
- 집합(aggregation): 어떤 클래스가 다른 클래스의 모임으로 구성
- 합성(composition): 포함된 클래스가 컨테이너 클래스가 없이는 존재할 수 없는 집합관계의 변형

UML 클래스 다이어그램

일반화 관계

Q 일반화(상속)

- 부모를 향한 화살표로 표시되는 하향 계층 관계
- 선/화살표는 부모 클래스의 종류에 따라 다름

❖ 클래스:

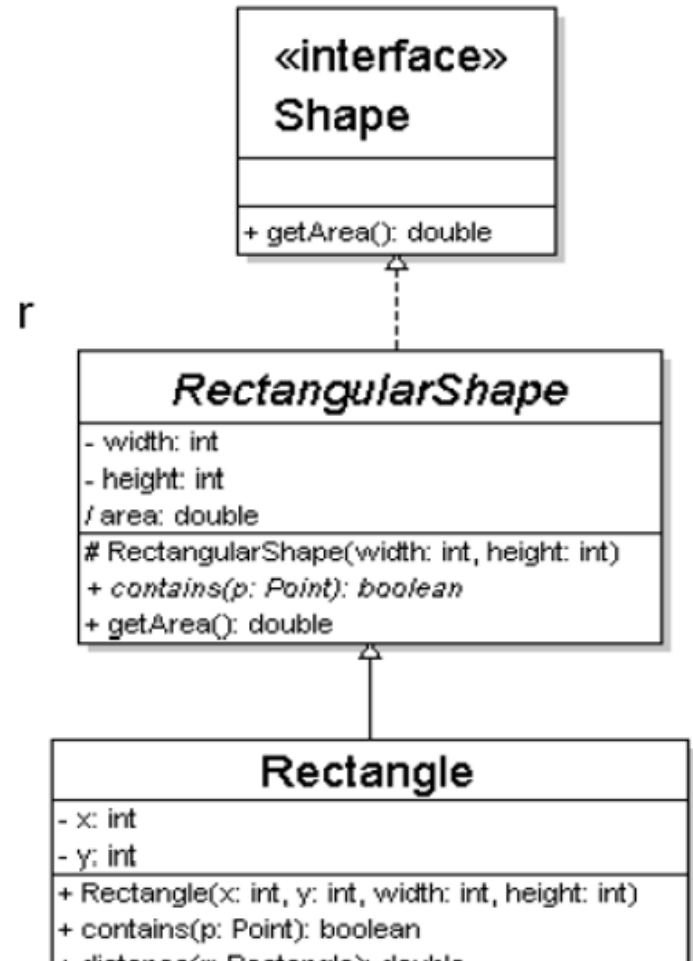
실선/검은 헤드 화살표

❖ 추상 클래스:

실선/흰 헤드 화살표

❖ 인터페이스:

점선/흰 헤드 화살표



UML 클래스 다이어그램

연관 관계

연관(association): 어떤 클래스의 인스턴스가 작업을 수행하기 위하여 다른 클래스를 알아야 하는 함

1. 다중도(multiplicity)

- * ☞ 0, 1, or more
- 1 ☞ 정확히 1개
- 2..4 ☞ 2개 내지 4개
- 3..* ☞ 3개 이상

2. 이름 – 객체들의 관계 이름

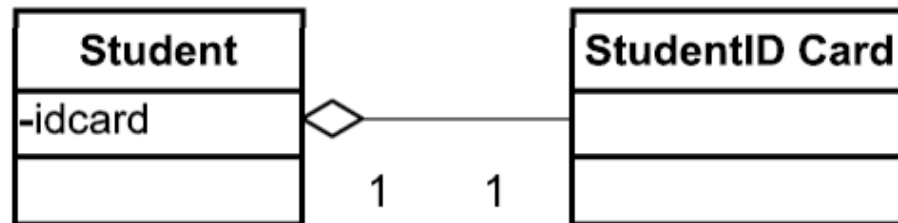
3. 방향성(navigability) – 질의의 방향, 객체 사이의 선으로 표시하며 양쪽 방향인 경우는 화살표시 없음

UML 클래스 다이어그램

연관 관계의 다중도

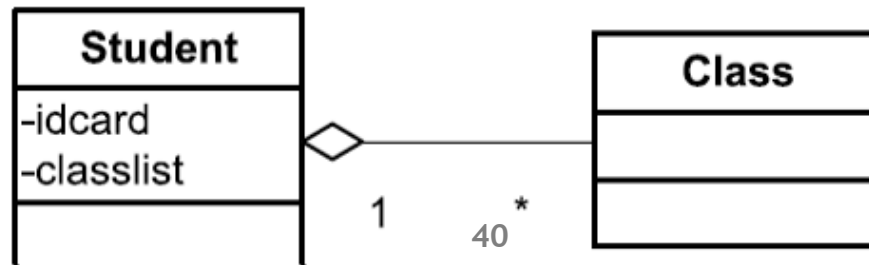
Q 1 대 1

- 학생 1명이 학생증(id card) 한 개만을 가진다.

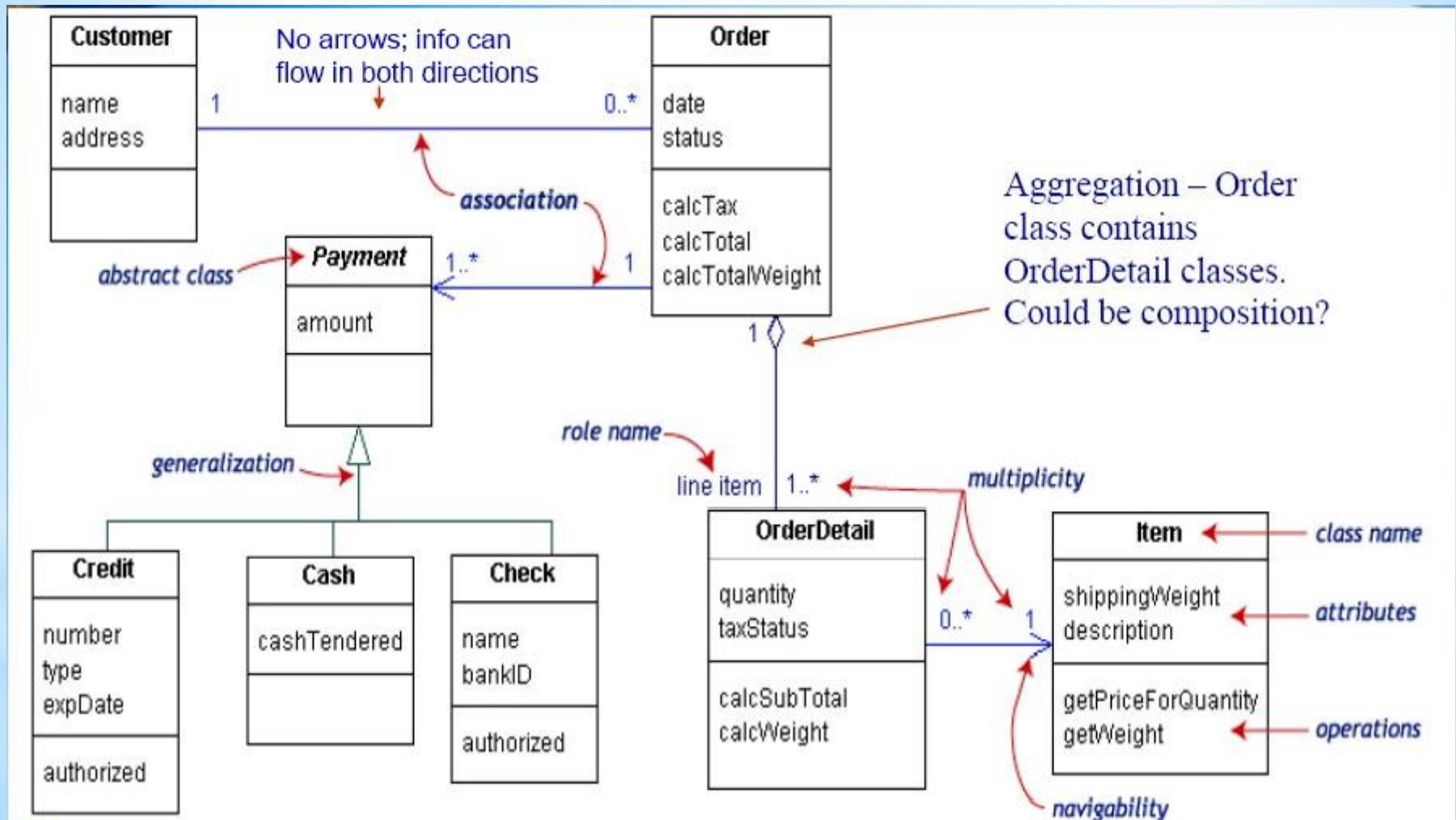


Q 1 대 다

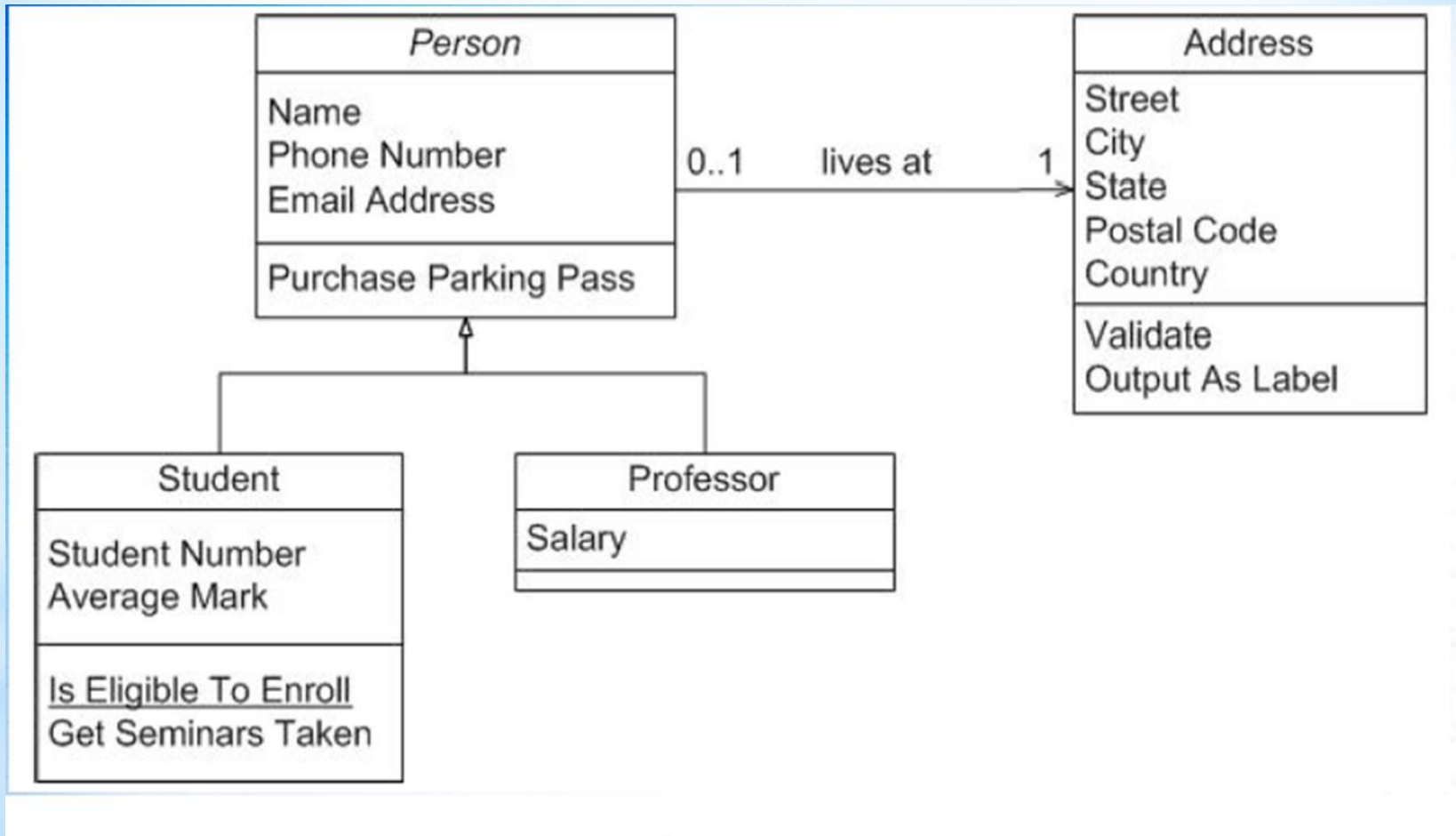
- 학생 1명이 여러 클래스를 수강할 수 있다.



클래스 다이어그램의 예



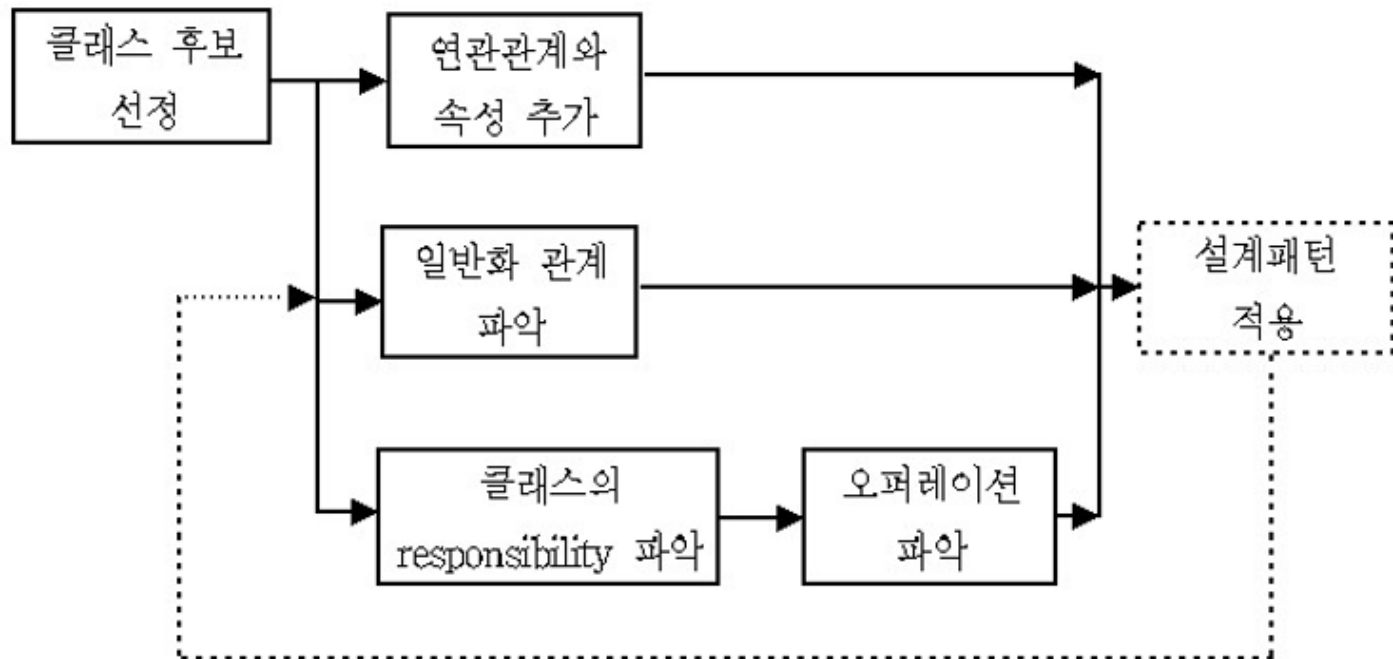
클래스 다이어그램의 예



클래스 다이어그램 작성 과정

Q 반복, 점증적 방법

○ 초벌로 작성 후 계속 추가, 삭제



캡스톤디자인 – 4주차 목차

1. 다이어그램
 - **Diagram**, 유스케이스, **UML** - 안용학교수
2. 캡스톤디자인 진행
 - 유스케이스, **Diagram** 작성
3. 팀별 토의 및 상담

감사합니다....

한동일 교수

- 연구실: 대양AI센터 721호. 02-3408-3751
- e-mail : dihan@sejong.ac.kr

안용학 교수

- 연구실: 대양AI센터 613호. 02-3408-3837
- e-mail : yohan@sejong.ac.kr