캡스톤디자인 – 6주차 목차

1. **UML / Sequence Diagram**
2. 캡스톤디자인 팀별 발표 (요구사항분석/유스케이스)
3. 팀별 토의 및 상담

V. UML 클래스 다이어그램

UML(Unified Modeling Language)

- **분석, 설계를 비주얼 화, 문서화 하기 위한 그래픽 언어**
- **Unified**
  - 이전의 OO 방법들의 통합
- **Modeling**
  - 객체지향 분석 설계를 위한 비주얼 모델링
- **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)

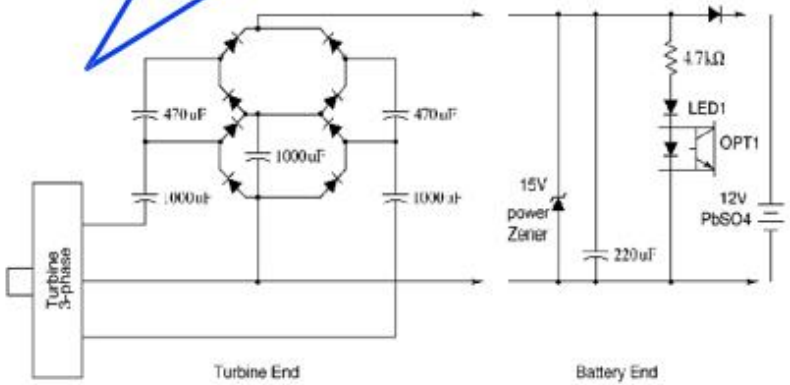
UML은 ( ) 이 아니다.

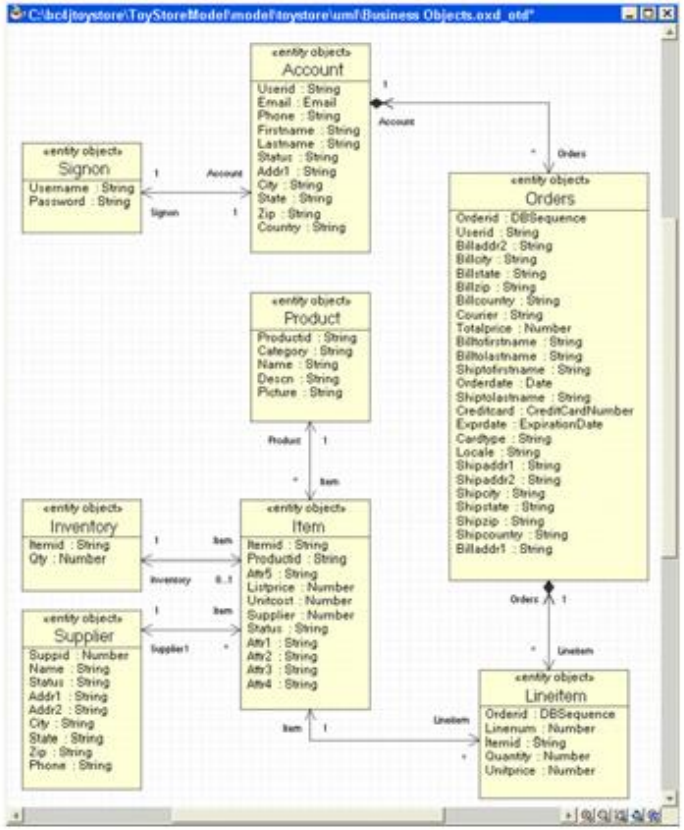
- Q 비주얼 프로그래밍 언어 환경
- Q 데이터베이스 표현 도구
- Q 개발 프로세스(SDLC)
- Q 모든 문제의 해결책
- Q 품질 보증 방안

UML(Unified Modeling Language)

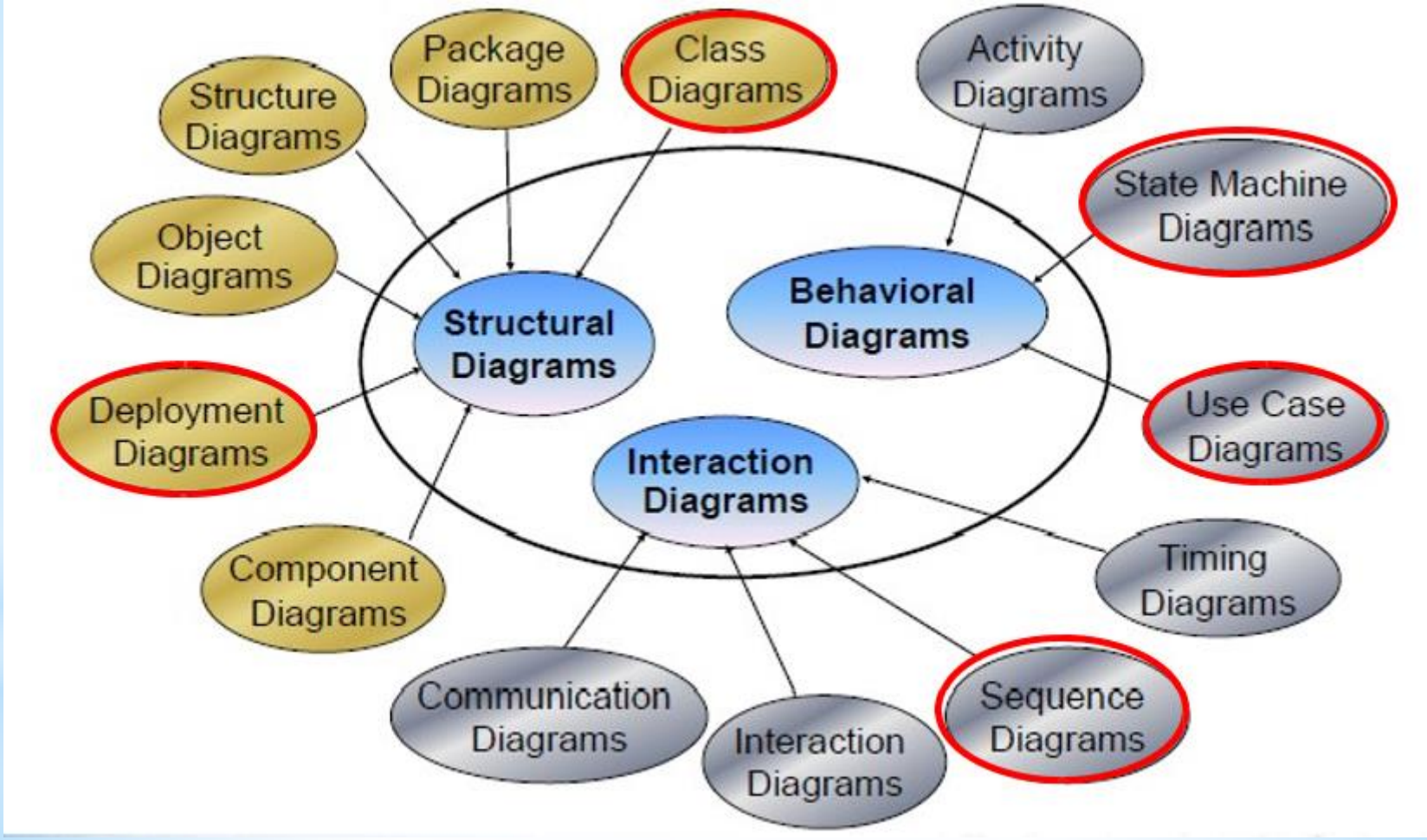
Every s/w engineer WILL understand UML diagrams.

Every h/w engineer understands curcuit 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

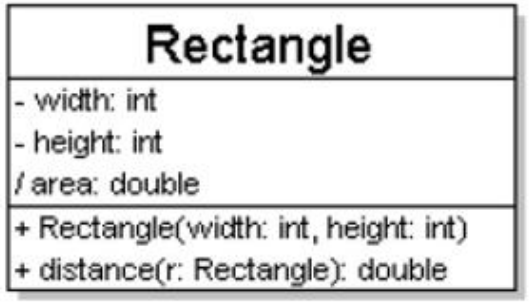
UML 클래스 다이어그램

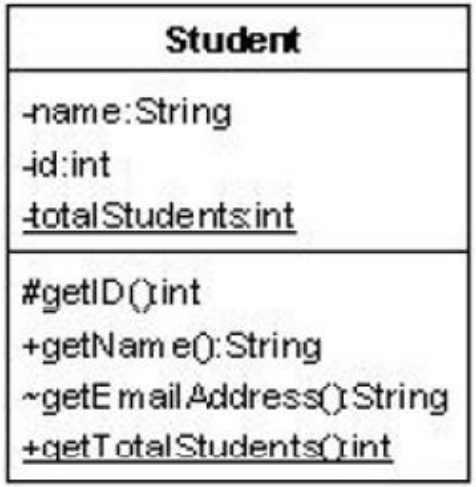
클래스 나타내기

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

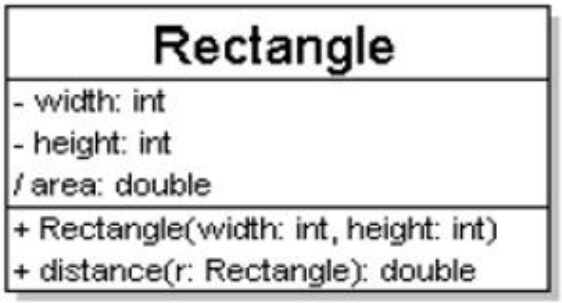
클래스 속성

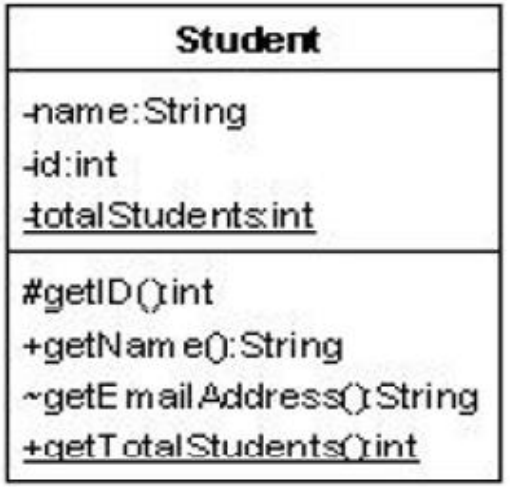
Q 속성(필드, 인스턴스 변수)

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

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

- 파생된 속성: 저장되지 않고 다른 속성값으부터 계산됨

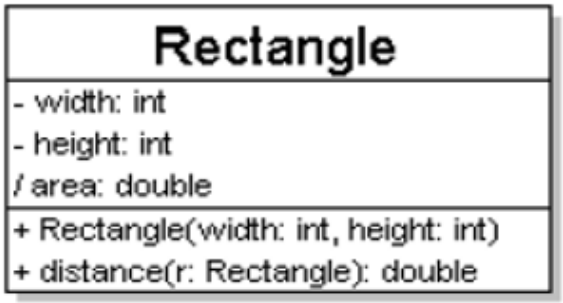


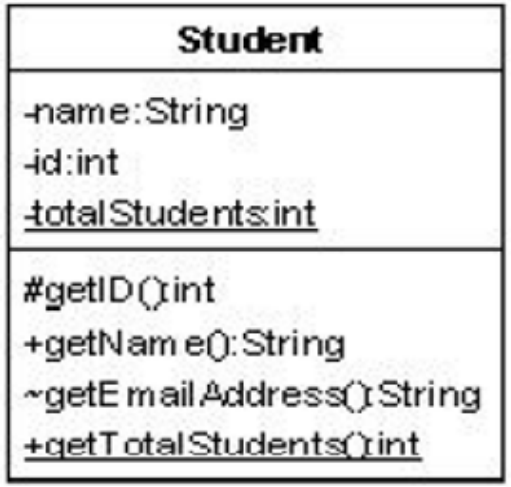


UML 클래스 다이어그램

클래스 오퍼레이션/메소드

- **오퍼레이션/메소드**
  - visibility name(prameters) : *return_type*
  - visibility: + public protecte
    # d private pack
    – age(디폴트)
    ~
  - Underline static method
  - 파리메타 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)

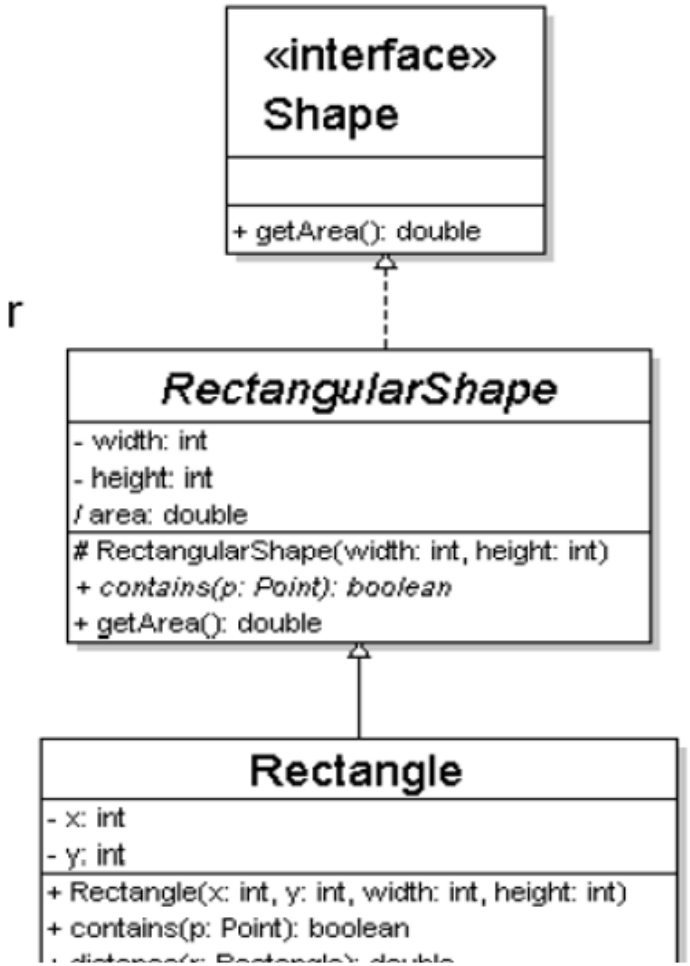
- 의존
- 집합(aggregation): 어떤 클래스가 다른 클래스의 모임으로 구성
- 합성(composition): 포함된 클래스가 컨테이너 클래스가 없이는 존재할 수 없는 집합관계의 변형

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

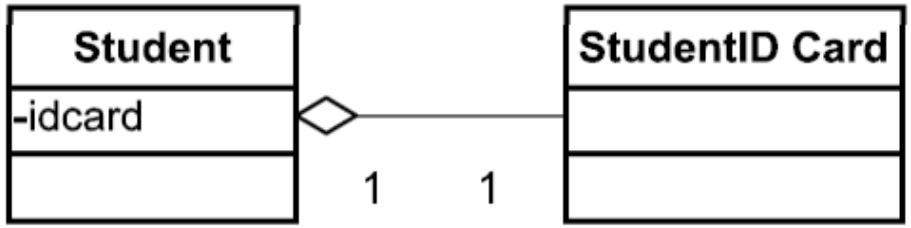
1. **다중도(multiplicity)**
   - * Đ 0, 1, or more
   - 1 Đ 정확히 1개
   - 2..4 Đ 2개 내지 4개
   - 3.. * Đ 3개 이상
2. **이름 – 객체들의 관계 이름**
3. **방향성(navigability) – 질의의 방향, 객체 사이의 선으로 표시하며 양쪽 방향인 경우는 화살표시 없음**

UML 클래스 다이어그램

연관 관계의 다중도

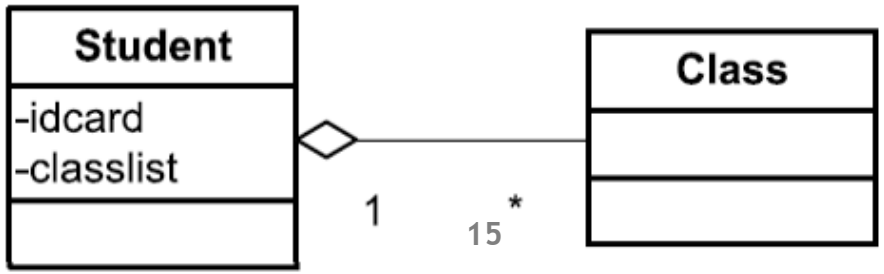
Q 1 대 1

❍ 학생 1명이 학생증(id card) 한 개만을 가진다.

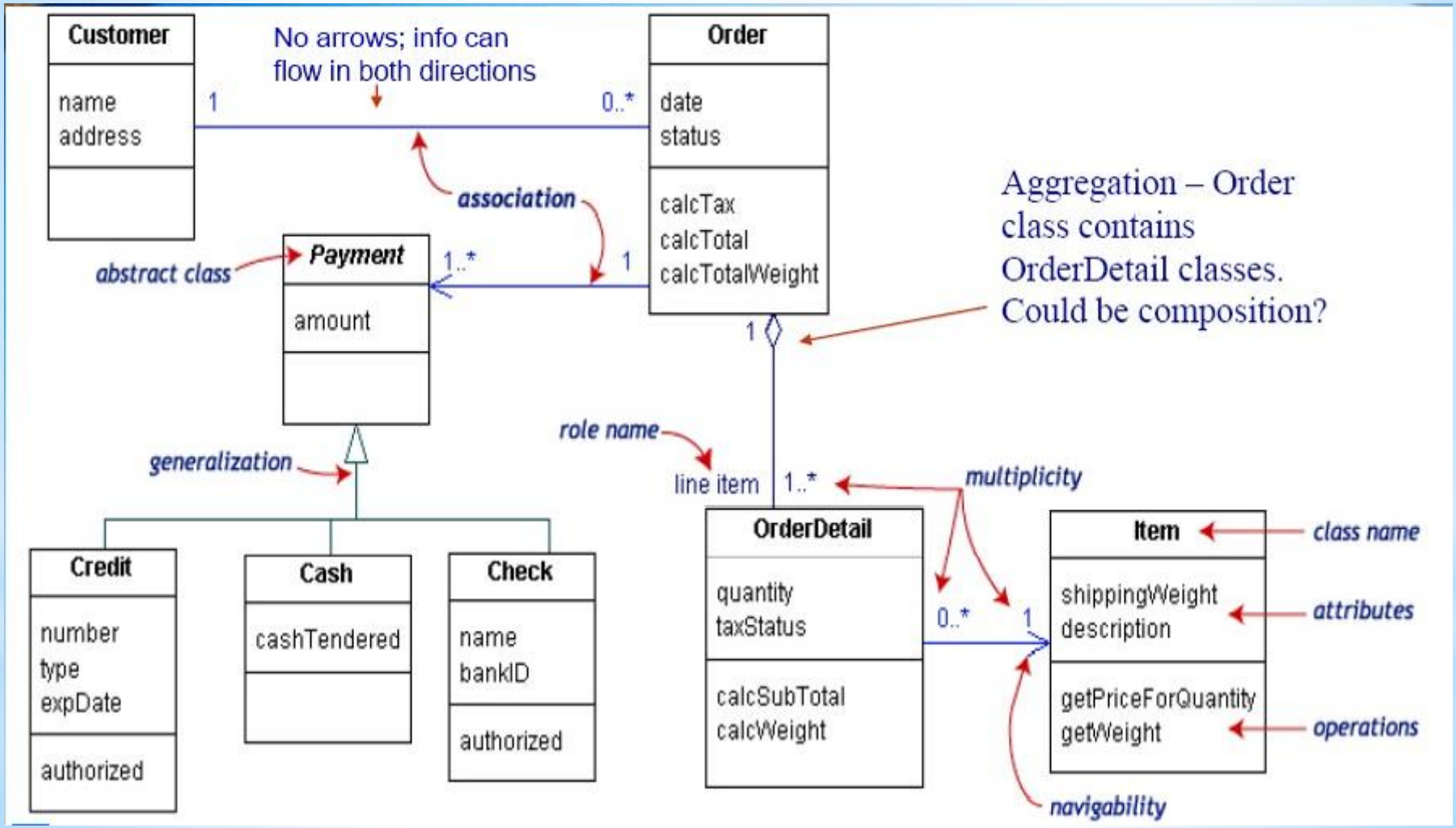


Q 1 대 다

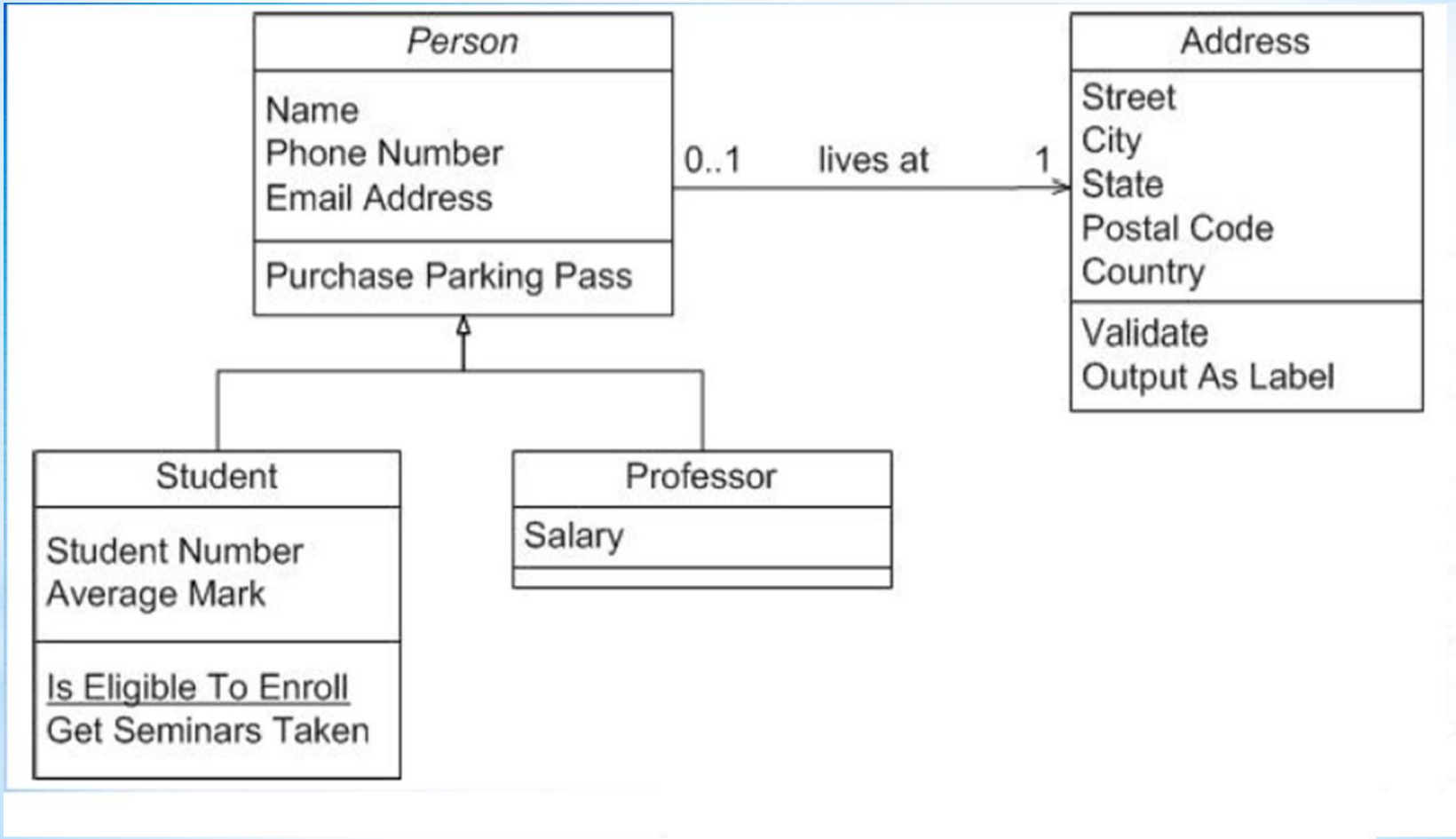
❍ 학생 1명이 여러 클래스를 수강할 수 있다.



클래스 다이어그램의 예


Customer
name
address
No arrows; info can flow in both directions
1
0..*
Order
date
status
calcTax
calcTotal
calcTotalWeight
association
abstract class
Payment
amount
1..*
1
Aggregation – Order class contains OrderDetail classes. Could be composition?
1
role name
line item
1..*
multiplicity
generalization
Credit
number
type
expDate
authorized
Cash
cashTendered
Check
name
bankID
authorized
OrderDetail
quantity
taxStatus
calcSubTotal
calcWeight
0..*
1
Item
shippingWeight
description
getPriceForQuantity
getWeight
class name
attributes
operations
navigability

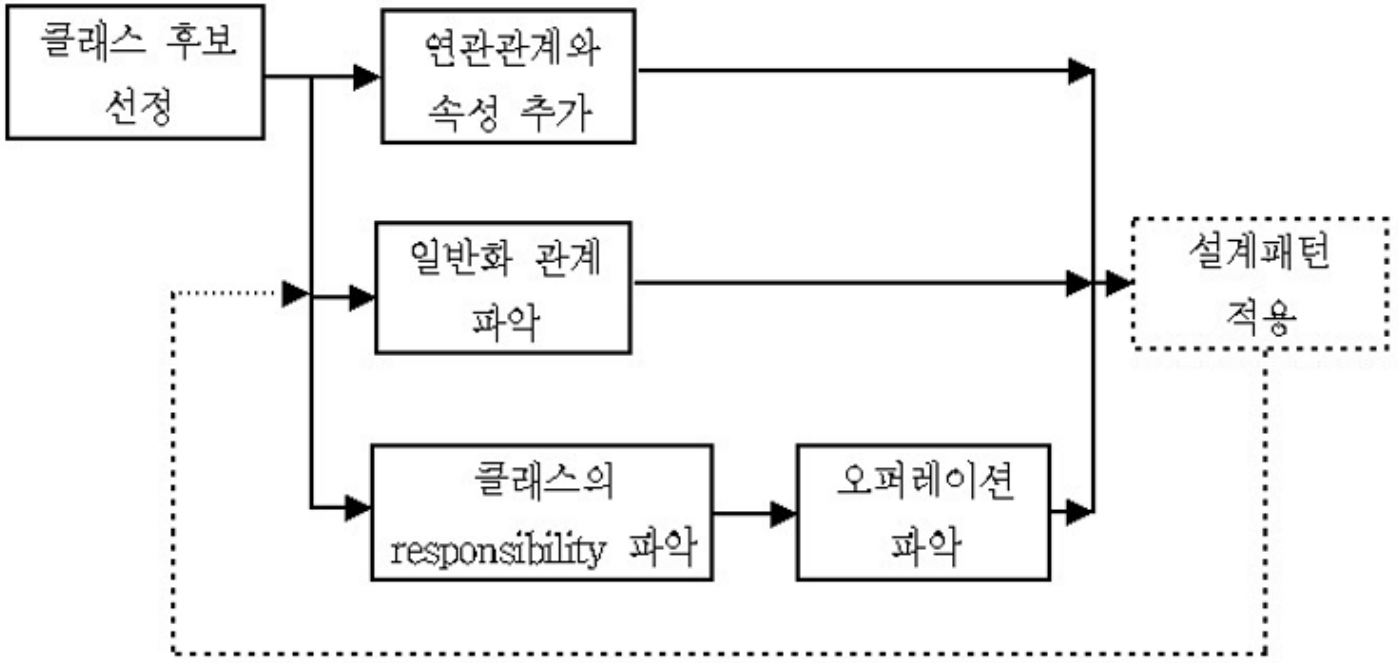
클래스 다이어그램의 예


Person
Name
Phone Number
Email Address
Purchase Parking Pass
0..1
lives at
1
Address
Street
City
State
Postal Code
Country
Validate
Output As Label
Student
Student Number
Average Mark
Is Eligible To Enroll
Get Seminars Taken
Professor
Salary

클래스 다이어그램 작성 과정

Q 반복, 점증적 방법

❍ 초벌로 작성 후 계속 추가, 삭제



감사합니다....

양효식 교수

- 연구실: 대양AI센터 808호. 02-3408-3840
- e-mail : hsyang@sejong.ac.kr

안용학 교수

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