

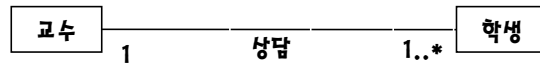
3. 클래스 사이의 다중성

다중성(multiplicity)은 하나의 객체에 연관된 다른 쪽 클래스의 가능한 객체 수를 나타낸다.

〈다중성(multiplicity)〉

1	1개의 객체(생략 가능)
0..1	0개 또는 1개
*	0개 이상(0은 제외)
1..*	1개 이상(0은 제외)
2..*	2개 이상
1,5..7	1 또는 5 또는 6 또는 7
m..n	최소 m개, 최대 n개

[예] 연관(association)



- 교수는 1명 이상의 학생을 상담한다.
- 학생은 1명의 교수에게 상담한다.
- 연관은 다중성을 가진다.

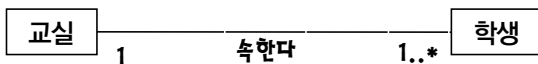
- 다중성을 표시하지 않은 것은 1을 의미한다.(다중성 1은 생략 가능)

〈다중성 표기하는 방법〉



- 다중성은 하나의 객체에 연관된 다른 쪽 클래스의 가능한 객체 수를 의미한다.
- 사람은 0개 이상(0..*)의 회사에서 일할 수 있다.
- 회사는 1명 이상(1..*)의 사람(직원)이 일한다.

[예제 1] 양방향 연관(association)

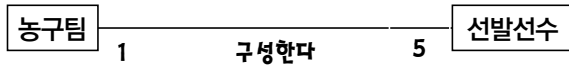


〈양방향 연관〉

- 교실에는 1명 이상의 학생이 존재하고.(학생 1..*)
- 학생은 하나의 교실에 존재할 수 있다.(교실 1)

2 <http://cafe.daum.net/pass365>(홍재연)

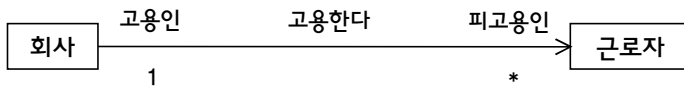
[예제 2] 양방향 연관(association)



〈양방향 연관〉

- 농구팀은 5명의 선발선수로 구성된다.(화살표를 표시하지 않는다)

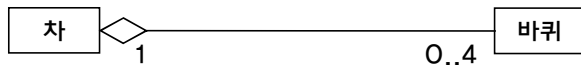
[예제 3] 단방향 연관(association)



〈단방향 연관〉

- 한 회사는 0명 이상의 근로자를 고용한다.(단방향 연관은 화살표를 표시한다)

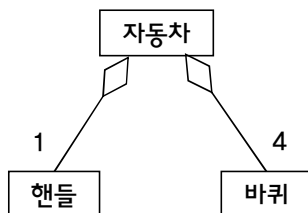
[예제 4] 집합(aggregation)



〈특수 연관〉

- 차(car)는 최대 4개의 바퀴(wheel)를 가질 수 있다.(바퀴 0..4)
- 바퀴(wheel)는 하나의 차(car)에만 장착할 수 있다.(차 1)
- 하지만, 차를 폐차했을 때, 바퀴는 독립적으로 다른 차에 장착할 수 있다.
- 차와 바퀴는 생명주기가 일치하지 않는다.

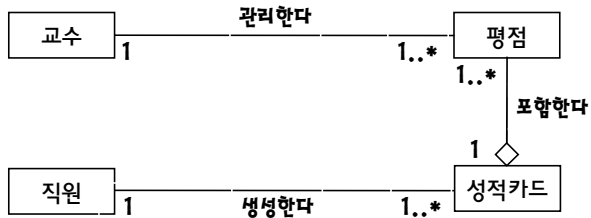
[예제 5] 집합(aggregation)



〈특수 연관〉

- 자동차는 핸들은 1개, 바퀴는 4개를 가진다.

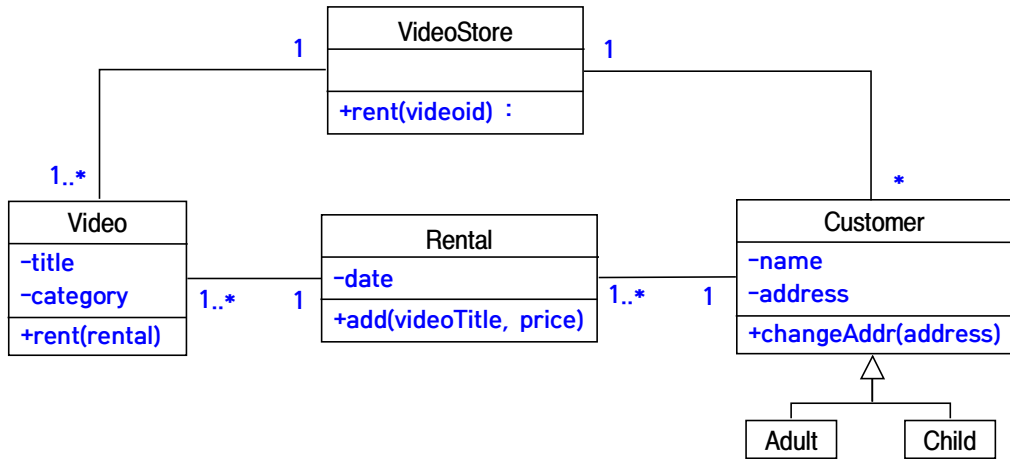
[예제 6] 성적관리



- ‘교수’는 1개 이상의 ‘평점’을 관리한다.(**평점 1..***)
- ‘평점’은 1명의 ‘교수’가 관리한다.(**교수 1**)
- ‘평점’은 하나의 ‘성적카드’에 포함된다.(**성적카드 1**)
 - ‘성적카드’에는 1개 이상의 ‘평점’이 포함된다.(집합)
- ‘성적카드’는 1개 이상의 ‘평점’을 포함한다.(**평점 1..***)
- ‘성적카드’는 1명의 ‘직원’에 의해서만 생성된다.(**직원 1**)
- ‘직원’은 1개 이상의 ‘성적카드’를 생성한다.(**성적카드 1..***)

기출문제 분석

1. 다음은 VideoRental 시스템의 도메인 모델을 Class Diagram으로 도식한 결과이다. 이에 대한 해석으로 가장 옳지 않은 것은? [2022년 군무원 7급]



- ① ‘Adult’와 ‘Child’ 모두 ‘name’, ‘address’라는 속성과 ‘changeAddr(address)’라는 메소드를 갖는다.
- ② ‘Rental’ 객체는 하나 이상의 ‘Video’ 객체와 연관이 있어야 하며, ‘Video’ 객체는 반드시 하나의 ‘Rental’ 객체와 연관이 있어야 한다.
- ③ ‘VideoStore’ 객체는 ‘Customer’ 객체가 1개 이상 연관되어 있어야 한다.
- ④ ‘Customer’ 객체는 하나 이상의 ‘Rental’ 객체와 연관이 있다.

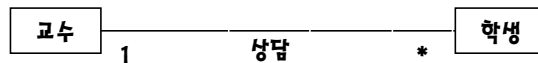
☞ 클래스 다이어그램

· ‘VideoStore’ 객체는 ‘Customer’ 객체가 1개 이상 연관되어 있어야 한다.(x) → 0개 이상(*)

// 다중성(multiplicity)

1	1개의 객체(생략 가능)
0..1	0개 또는 1개
*	0개 이상(0..*)
1..*	1개 이상(0은 제외)
2..*	2개 이상
1,5..7	1 또는 5 또는 6 또는 7
m..n	최소 m개, 최대 n개

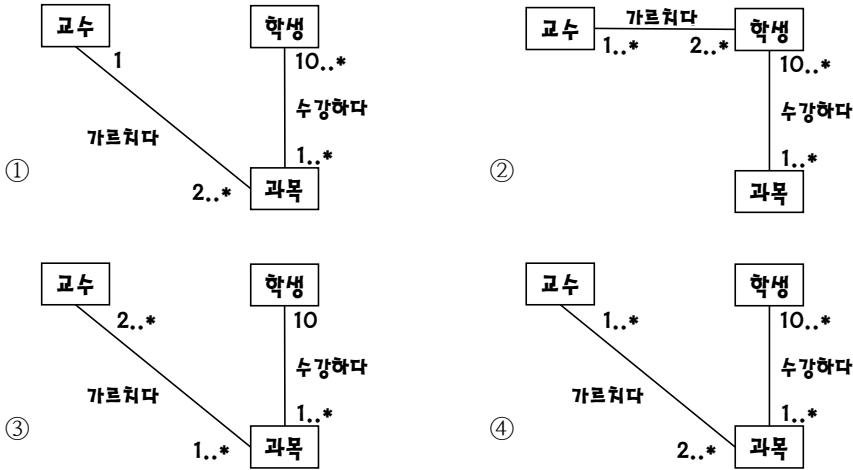
[예] 연관(association)



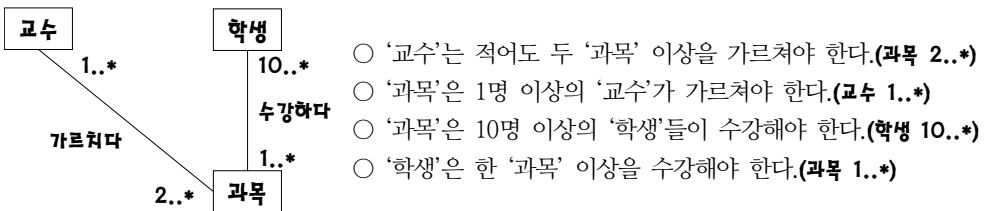
- 교수는 0명 이상의 학생을 상담한다.
- 학생은 1명의 교수에게 상담한다.
- 연관은 다중성을 가진다.

2. 다음 사항을 UML 클래스 다이어그램으로 가장 잘 나타낸 것은? [2012년 국가 7급]

- '교수'는 적어도 두 '과목' 이상을 가르쳐야 한다.
- '과목'은 한 명 이상의 '교수'가 가르쳐야 한다.
- '과목'은 열 명 이상의 '학생'들이 수강해야 한다.
- '학생'은 한 '과목' 이상을 수강해야 한다.



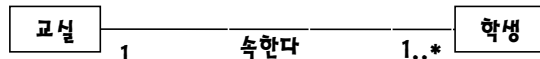
☞ UML 클래스 다이어그램에서 연관 관계의 다중성



◎ 다중성(multiplicity)

1	객체가 1개(생략 가능)
0..1	0개 또는 1개
*	0개 이상(0..*)
1..*	1개 이상(0은 제외)
2..*	2개 이상
m..n	최소 m개, 최대 n개

[예] 연관(association)

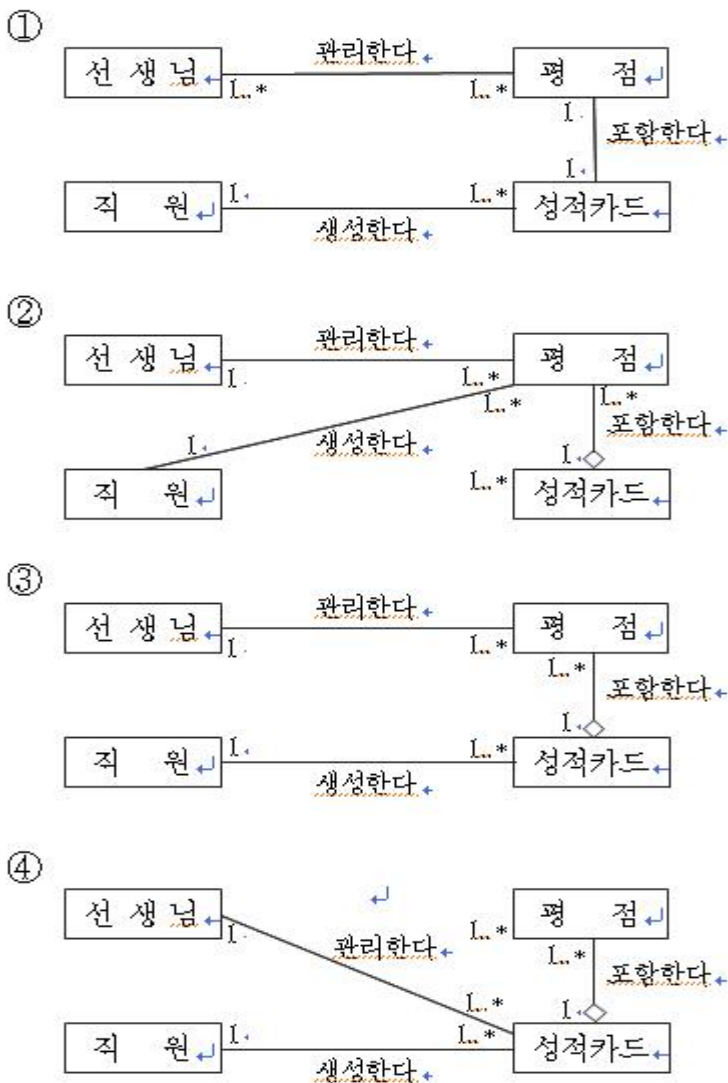


- 교실에는 1명 이상의 학생이 존재하고.
- 학생은 하나의 교실에 존재한다.

· 다중성은 하나의 객체에 연관된 다른 쪽 클래스의 가능한 객체 수를 의미한다.

3. 다음 사항을 가장 잘 표현한 클래스 다이어그램은? [2007년 국가 7급]

- '선생님'은 적어도 하나의 '평점'을 관리한다.
- '평점'은 한분의 '선생님'으로부터 관리된다.
- '평점'은 하나의 '성적카드'에 포함된다.
- '성적카드'는 적어도 하나 이상의 '평점'을 포함한다.
- '직원'은 적어도 하나 이상의 '성적카드'를 생성한다.
- '성적카드'는 한 사람의 '직원'에 의해서만 생성된다.



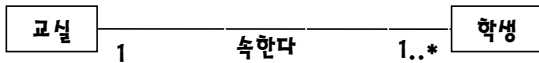
☞ 클래스 다이어그램에서 연관 관계의 다중성

● 다중성(multiplicity)

1	객체가 1개(생략 가능)
0..1	0개 또는 1개
*	0개 이상(0..*)
1..*	1개 이상(0은 제외)
m..n	최소 m개, 최대 n개

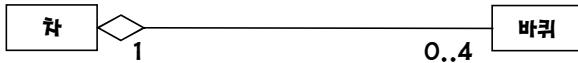
다중성은 하나의 객체에 연관된 다른 쪽 클래스의 가능한 객체 수를 의미한다.

[예 1] 연관(association)



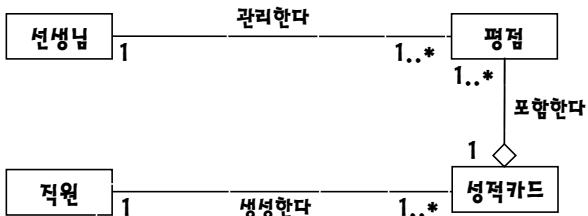
- 교실에는 1명 이상의 학생이 존재하고.(학생 1..*)
- 학생은 하나의 교실에 존재한다.(교실 1)

[예 2] 집합(aggregation)



- 차(car)는 최대 4개의 바퀴(wheel)를 가질 수 있다.(바퀴 0..4)
- 바퀴(wheel)는 하나의 차(car)에만 장착할 수 있다.(차 1)
→ 하지만, 차를 폐차했을 때, 바퀴는 독립적으로 다른 차에 장착할 수 있다.
- 차와 바퀴는 생명주기가 일치하지 않는다.

● 주어진 문제 분석



- '선생님'은 1개 이상의 '평점'을 관리한다.(평점 1..*)
- '평점'은 1명의 '선생님'에 의해 관리된다.(선생님 1)
- '평점'은 하나의 '성적카드'에 포함된다.(성적카드 1)
→ '성적카드'에는 다수의 '평점'이 포함된다.(집합)
- '성적카드'는 1개 이상의 '평점'을 포함한다.(평점 1..*)
- '직원'은 1개 이상의 '성적카드'를 생성한다.(성적카드 1..*)
- '성적카드'는 1명의 '직원'에 의해서만 생성된다.(직원 1)

4. UML의 클래스 다이어그램에서 클래스 사이의 관계에 대한 설명으로 옳지 않은 것은? [2021년 계리]

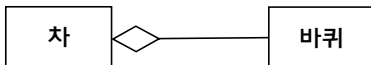
- ① 일반화(generalization) 관계는 일반화한 부모클래스와 실체화한 자식클래스 간의 상속 관계를 나타낸다.
- ② 연관(association) 관계에서 다중성(multiplicity)은 관계 사이에 개입하는 클래스의 인스턴스 개수를 의미한다.
- ③ 의존(dependency) 관계는 한 클래스가 다른 클래스를 참조하는 것으로 지역변수, 매개변수 등을 일시적으로 사용하는 관계이다.
- ④ 집합(aggregation) 관계는 강한 전체와 부분의 클래스 관계이므로 전체 객체가 소멸되면 부분 객체도 소멸된다.

☞ UML의 클래스 다이어그램

- 집합(aggregation) 관계는 강한 전체와 부분의 클래스 관계이므로 전체 객체가 소멸되면 부분 객체도 소멸된다.(x) → 집합은 약한 전체 부분관계이다.

집합 (aggregation)	<ul style="list-style-type: none"> • 약한 전체 부분관계 • 집합은 생명주기가 일치하지 않는다. • 부분은 자체적으로 존재할 수 있고, 없어질 수도 있다. • 집합은 집단화라고도 한다. • 그리는 방법 : 포함하고 있는 클래스 쪽에 끝이 빈 다이아몬드로 표시 • 예 : 회사와 부서의 관계 - 회사에는 여러 부서가 존재한다.
합성[복합] (composition)	<ul style="list-style-type: none"> • 강한 전체 부분관계 • 합성은 생명주기를 공유한다. • 부분만으로는 스스로 존재할 수 없다. • 합성은 집합 관계보다 더 끈끈한 관계이다.(더 강한 집합) • 그리는 방법 : 포함하고 있는 클래스 쪽에 검은 다이아몬드로 표시 • 예 : 손(hand)은 손가락을 포함하고 있다.

- 집합(aggregation) : 차와 바퀴의 관계(생명주기가 불일치하는 경우)



- 합성(composition) : 손과 손가락의 관계(생명주기가 일치하는 경우)

