

객체관계 :

- 상속 (일반상속, 인터페이스)
- 포함
- 사용

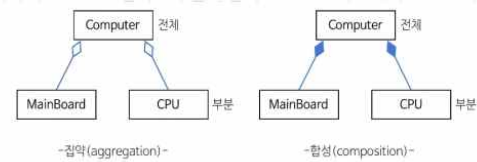
1. 상속

◆ 상속에서의 결합도 비교

구분	강한 결합 (Strong Coupling) - 직접 상속	약한 결합 (Loose Coupling) - 인터페이스
유연성	낮음 (부모 변경 시 자식 영향)	높음 (인터페이스만 유지하면 변경 가능)
확장성	어려움 (새 클래스 추가 시 기존 코드 변경 필요)	쉬움 (새 클래스 추가해도 기존 코드 영향 적음)
의존성	부모 클래스에 강하게 의존	인터페이스에 의존 (구현과 분리)
예제	<code>class Dog extends Animal</code>	<code>class Dog implements SoundMaker</code>

2. has a

(실선화살표 , 채워진 다이아몬드, 빈 다이아몬드)



◆ 강한 결합 vs 약한 결합 비교

구분	강한 결합 (Strong Coupling)	약한 결합 (Loose Coupling)
객체 생성	<code>new</code> 로 직접 생성	외부에서 주입 (DI)
유연성	낮음	높음
의존성	특정 클래스에 강하게 의존	인터페이스나 DI 사용으로 의존 감소
테스트 용이성	어려움 (Mocking 어려움)	쉬움 (Mock 객체 사용 가능)
코드 변경 영향	변경 시 큰 영향	변경 시 영향 적음

3. 사용(의존관계) (점선 화살표)

◆ 의존 관계 vs 연관 관계 비교

구분	의존 관계 (Dependency)	연관 관계 (Association)
지속성	일시적 (메서드 호출 시)	지속적 (멤버 변수로 저장)
강도	약한 관계	강한 관계
UML 표현	점선 화살표(→)	실선 화살표(→)
예시	매개변수로 객체 받기	필드로 객체 저장
예제 코드	<code>void print(Document doc)</code>	<code>private Document doc;</code>