

## 0311 정리

- 상속 : 필드와 메소드를 모두 물려받는다
  - `public void 자식클래스 extends 부모클래스(){...}`
  - `private`으로 생성된 필드는 직접 접근할 수 없다.
  - `getter / setter`으로 접근 할 수 있지만 비효율적이다
  - `super`을 이용하면 조상 객체에 접근하면 된다
  - 자바는 다중상속을 지원하지 않는다.
  - 부모클래스의 생성자, 초기화 블록은 상속 안됨.
  - 모든 클래스는 결국 `Object`의 후손이다
- 상속의 장점
  - 코드의 재사용성 감소
  - 코드의 길이 감소
  - 코드의 유지보수 용이함
- `super()` : 부모클래스로부터 상속받은 필드나 메소드를 자식 클래스에서 참조하는데 사용한다.
  - 인스턴스 변수(`new`)로 생성된 객체 )의 이름이 같을 경우 `this.`으로 구분한다.
  - 반드시 생성자 최상단에 작성되어야 한다.

?

- `super.` : 자식 클래스에서 부모클래스에 접근해서 필드나 메소드를 참고하는데 사용한다..



- **getter / setter** 을 사용하지 않고 **Supur**을 이용해서 값을 한번에 전달한다.

**Overriding:**부모에게 받은 메소드를 자식의 필요성에 따라 **재정의**하는 것

오버라이딩의 성립 조건

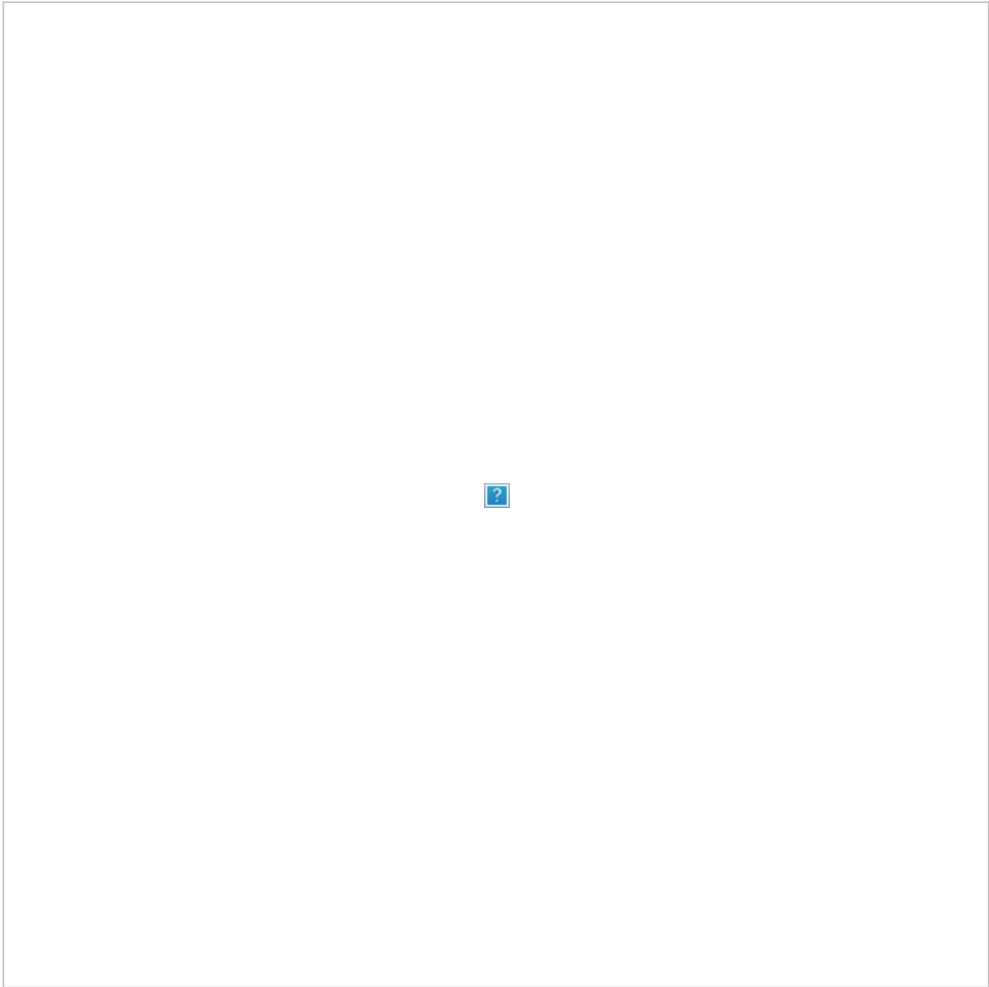
1. 메소드의 이름이 같아야한다.
2. 메소드와 반환형이 같아야한다.
3. 매개변수가 있는 경우 같아야한다.
4. 접근제한자가 같거나 더 넓은 범위이어야 한다.
5. 예외처리 범위는 같거나 더 좁은 범위이어야 한다.

**@overriding**을 할 메소드 위에 **@Override**를 명시한다.



**@부모의 private** 메서드는 **overriding** 불가

오버라이딩 오버로딩 아예 다른 것이다



오버로딩은 매개변수의 개수 / 타입 /순서를 제외하고는 신경쓰지 않는다  
둘이 다른 것이지만 헷갈리는 경우가 상당히 많음으로 주의하자.

final 의 활용

- final \_ 클래스
  - 선언시 상속 불가
- final \_ 메소드
  - 선언시 오버라이딩 불가

다형성(Polymorphsim)

- 상속을 이용한 기술로 부모타입 참조변수로 자식타입 객체를 다루는 것
- 조상 클래스의 참조 변수로 자식 클래스의 객체를 다룰 수 있는 특징

2가지 장점

<sup>1.</sup>업 캐스팅( Up Casting)

- 상속 관계에 있는 부모 ,자식클래스 간에 부모 타입의 참조형 변수가 모든 자식 타입의 객체 주소를 참조 할 수 있음
- ex)



다운 캐스팅(Down Casting)

- 자식객체의 주소를 받은 부모참조형 변수를 가지고 자식의 멤버를 참조해야 할 경우 부모클래스 타입의 참조형 변수를 자식 클래스 타입으로 강제 형변환 하는 것

다형성의 활용

객체배열

매개변수

바인딩