

상속(Inheritance): 객체 지향 프로그램에서 부모클래스의 '필드' 와 메소드를 자식 클래스에게 물려주는 것

상속의 장점: 1. 이미 개발된 클래스를 재사용해서 새로운 클래스를 만들기 때문에 중복 코드를 줄일 수 있음.

2. 클래스의 수정을 최소화 할수 있음 (부모 클래스를 수정하면 모든 자식클래스에 수정효과를 가져옴.)

클래스 상속: 자식클래스가 부모 클래스를 선택한다. (extend를 이용하여 부모 클래스를 정할수 있음.)

부모 생성자 호출: 자바에서는 자식 객체를 생성하면 부모 객체가 먼저 생성된 다음에 자식 클래스가 생성됨.

예 (자식클래스 변수 = new 자식클래스 ();

모든 객체는 생성자를 생성해야하는데 부모 객체는 자식생성자를 선언하면 !!!!!super()!!!!!! 가

컴파일 과정에서 자동 추가 된다. super()는 부모의 기본 생성자를 호출 한다.

만약 부모클래스에 기본 생성자가 없으면 자식 생성자 선언못함.

만약 부모클래스에 기본 생성자 x 매개변수를 갖는 생성자만 있다면

매개값의 타입과 개수가 일치하는 부모생성자를 호출함.

메소드 오버라이딩: 상속된 메소드를 자식클래스에서 재 정의 하는 것. (메소드가 오버라이딩 되면

해당 부모 메소드는 숨겨지고 자식 메소드가 우선적 사용됨.

메소드 오버라이딩 할 때 주의사항

1. 부모 메소드의 선언부(리턴타입, 메소드 이름, 매개변수)와 동일해야함.

2. 접근제한을 더 강하게 할수 없음. (예: public -> private 로 변경불가)

3. 새로운 예외를 throws 할수 없다.

@Override : 정확히 오버라이딩이 되었는지 체크해주는 어노테이션.

@Override를 붙이면 정확히 체크하고 문제가 있다면 컴파일에러.

자식메소드 내에서 부모 메소드를 호출하려면 super 키워드와 도트(.)연산자를 사용하면 숨겨진 부모 메소드 호출 가능.

final 클래스: class 앞에 final을 붙이면 최종적인 클래스이므로 더 이상 상속할수 없다.

즉 부모클래스가 될수 없어 자식클래스를 만들 수 없음.

final 메소드: 최종적인 메소드 이므로 오버라이딩 할수 없음.(부모 클래스에 선언된 final 메소드는 자식에서 재정의 할수없음.)

protected 접근 제한자: 상속과 관련있고 public과 default의 중간 정도의 접근 제한이며

같은 패키지이거나 자식 객체만 사용가능.

(필드 생성자 메소드 선언에 사용됨.)

만약 D에서 접근하고 싶은 클래스가 다른 패키지에 있어도 A의 자식 클래스이므로

A의 PROTECTED 필드,생성자,메소드에 접근 가능. (NEW 연산자 사용 못함. 자식생성자에서

super()로 A 생성자 호출 가능.)