

● 선언 방법 ● 주요 문법

타입 변환 : 자동과 강제로 구분

자동 타입 변환 : `Cat cat = new Cat();`
`Animal animal = cat;` `Animal animal = new Cat();`

바로 위 부모가 아니라도 상위타입이라면 자동 타입 변환이 일어날 수 있다.

변환 후 부모 클래스에서 선언된 필드&메소드만 접근 가능

메소드가 @Override 되었으면 오버라이딩된 메소드가 호출됨.

강제 타입 변환 : `Child child = (Child) parent;`

부모타입으로 자동 타입 변환 후 다시 자식타입으로 강제 변환 사용 가능

자식 타입에 선언된 필드, 메소드를 사용해야한다면 사용.

다형성 :

사용방법은(메소드는) 동일 But 실행 결과(오버라이딩 내용) 다양.

매개변수 다형성 : 제공받는 자식객체의 종류에 따라 메소드의 실행결과가 달라지는 것

객체 타입 확인 : `A객체 instanceof B타입` / A객체가 B타입이면 true, 아니면 false

추상 클래스 :

`public abstract class 클래스명` / 클래스들의 공통 필드, 메소드를 추출해서 선언한 클래스

1. 객체를 직접 만들지 못함. 2. 상속을 통해 자식클래스만 만들 수 있음.

3. 생성자 반드시 있어야 함.(자식클래스의 super()를 통한 호출) 4. 자식 클래스에서 재정의 가능

봉인된 클래스 : `public sealed class 클래스명 permits 자식클래스1, 자식클래스2`

자식 클래스 : non-sealed (봉인 해제), final(더이상 상속 불가), sealed(봉인 클래스)

인터페이스 :

A객체와 구현객체 B를 연결하는 역할 / 구현객체 B가 C로 변경되었을 때 A를 수정하는 번거로움을 방지

인터페이스 선언 : `public interface 인터페이스명`

구현 객체 선언 : `public class 구현객체명 implements 인터페이스명`

인터페이스는 참조타입이므로 객체를 참조하지 않는다는 뜻의 null을 대입할 수 있다.

`RemoteControl rc = new Television();`

상수필드 :

인터페이스가 가질 수 있는 public static final의 특성을 갖는 불변의 상수 필드 / 구현객체와 관계 없음.

`int MAX_VOLUME = 10;` (대문자/단어 구분)

추상 메소드 : 인터페이스는 구현 클래스가 재정의해야 하는 추상클래스를 가진다.

컴파일 과정에서 자동으로 public abstract가 붙어 선언할 때 `void 메소드명();` 으로 선언 가능

재정의할 때 public 보다 낮은 접근 제한자를 사용할 수 없다.

디폴트 메소드 :

인터페이스 내에서 완전한 실행코드를 갖는 메소드. 실행부가 있으며 클래스 메소드 앞에 default를 붙여 선언

`default void 메소드명(매개변수)`

디폴트 메소드도 구현 클래스에서 재정의 가능 `public void 메소드명(매개변수) *default 생략`