클래스

- 멤버변수(부품,요소)와 멤버메서드(기능,동작)로 구성

- 생성자 -> 매개변수를 초기화

- 기본생성자 -> 생성자에 매개변수가 없는 경우

- 복사생성자 -> 매개변수가 같은 클래스 내의 객체인 생성자

static 멤버변수(클래스변수)와 일반(객체) 멤버변수

클래스 멤버변수 : 개체 생성 없이 클래스이름.멤버변수로 사용 가능

일반 멤버변수 : 객체를 생성하고 객체이름.멤버변수로 사용 가능

상속

- 상속을 받으면 부모 클래스에 있는 멤버변수와 멤버메서드를 물려받는다(재사용을 위해)

->단, 접근제한자가 private면 자식 클래스에서 접근할 수 없다

- 접근제한자가 public, protected일 때 상속 가능(default는 패키지가 다를 경우 사용 불가)

조건

- is a (~은 ~이다)관계가 성립하면 다른 클래스를 상속할 수 있다

->기아 자동차는 자동차다라는 말이 성립하면 기아자동차는 상속받을 수 있고 성립하지 않는다면 상속받을 수 없다

포함관계

- has a (~은 ~을 가지고 있다)관계가 성립하면 다른 클래스를 포함 가능

추상 클래스 - 추상 메서드를 포함한 클래스

- 추상 메서드 : 선언부만 있고 구현부는 없는 메서드

- 추상 클래스는 상속받은 자식 클래스에서만 구현할 수 있고, 상속받으면 반드시 오버라이딩을 해야 한다

인터페이스 - 추상메서드와 상수만을 멤버로 가질 수 있는 추상 클래스

사용 이유

- 클래스의 기능을 한눈에 볼 수 있다

- 여러 개발자가 동시에 개발하는 경우 표준화를 시킨다

클래스 - 아파트 설계도

추상클래스 - 화장실이 안그려진 미완성 아파트 설계도

인터페이스 - 방 개수, 화장실 개수, 거실 개수와 위치 정도만 그려진 아파트 설계도

상속 - is a (~는 ~이다)

포함 - has a (~는 ~를 가지고 있다)

클래스 상속은 단일 상속만 가능(부모 클래스는 1가지만 가능)

Object - 모든 부모 클래스의 조상 클래스

오버라이딩 - 부모 클래스의 메서드 재정의

super - 부모클래스, super() = 부모클래스의 생성자 호출

제어자

default - 같은 패키지 내에서만 사용 가능

private - 같은 클래스 내에서만 사용 가능

protected - 같은 패키지와 자식 클래스에서 사용 가능

public - 제한 없음

다형성 - 부모의 클래스를 이용해 상속받는 모든 자식 클래스에 일괄 작업

참조변수의 형변환

- 자손 -> 부모 : 자동

- 부모 -> 자손 : 명시적(조건적으로 가능)

가능한 경우

Shape s = new Rect(); //자식 클래스의 객체를 생성 후 부모 클래스에 대입(큰 것에서 작은 것으로)

Rect r = (Rect)s;

불가능한 경우

Shape s = new Shape();

Rect r = (Rect)s;

instanceof 연산자 - 형변환 가능 여부를 알려주는 연산자

매개변수의 다형성

- 매개변수를 잘 선정하여 다른 매개변수들을 동일하게 처리

추상클래스 - 추상메서드를 포함한 클래스

추상메서드 - 선언부만 있는 메서드

추상클래스를 상속받으면 반드시 추상메서드를 구현해야 한다

인터페이스 - 추상메서드와 상수(클래스 멤버변수)로만 이루어짐

인터페이스는 다중 상속이 가능(extends와 implements 함께 사용 가능)

인터페이스의 구현 - implements를 이용하여 인터페이스를 선언

interface A{}

class B implements A{

오버라이딩

}