자바의 객체지향 특성중 하나인 다형성이란 ? 여러가지 형태를 가질 수 있는 능력

-한 타입의 참조변수로 여러 타입의 객체를 참조할 수 있도록 해서 다형성을 프로그램적으로 구현하였다.

-조상클래스 타입의 참조변수로 자손클래스의 인스턴스를 참조할 수 있도록 하였다.

-참조변수가 사용할 수 있는 멤버의 개수는 인스턴스의 멤버 개수보다 같거나 적어야한다.

-즉, 자손타입의 참조변수로 조상타입의 인스턴스를 참조할 수는 없다.

참조변수의 형변환

-상속관계에 있는 클래스끼리 가능

-자손타입에서 조상타입(Up-casting) : 형변환 생략가능

-조상타입에서 자손타입(Down-casting) : 형변환 생략불가

--참조변수가 다룰 수 있는 멤버의 개수를 늘이는 것이라, 실제 인스턴스의 멤버개수보다 참조변수가 사용할 수 있는 멤버의 개수가 더 많아짐

-형변환을 수행하기 전에 instanceof연산자로 참조변수가 참조하는 실제 인스턴스의 타입을 확인하는 것이 안전함(어떤 타입에 대한 instanceof 연산의 결과가 true이면 검사한 타입으로 형변환이 가능하다는 것)

ex) Tv t = new CaptionTv();

=Tv t = (Tv)new CaptionTv();

=CaptionTv c = new CaptionTv();

Tv t = (Tv)c;

FireEngine과 Ambulance가 각각 Car 클래스를 부모클래스로 두고 상속관계를 맺고있음

Car car = null;

FireEngine fe = new FireEngine();

FireEngine fe2 = null;

Car = fe; //car = (Car)fe; 에서 형변환 생략. 업캐스팅

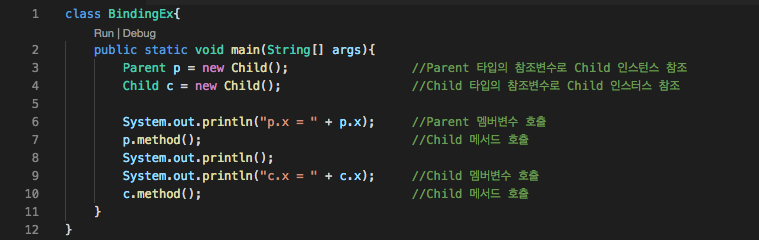
Fe2 = (FireEngine)car; //형변환 생략 불가. 다운캐스팅

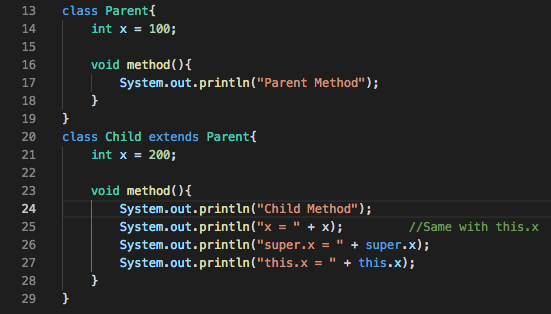
참조변수와 인스턴스의 연결

-조상클래스에 선언된 “멤버변수”와 같은 이름의 인스턴스 변수를 자손클래스에 중복으로 정의했을 때 -- 조상타입의 참조변수로 자손 인스턴스를 참조하는 경우와, 자손타입의 참조변수로 자손인스턴스를 참조하는 경우는 서로 다른 결과를 얻음.

-즉, 멤버변수는 참조변수의 타입에 따라 달라짐

-메서드 같은 경우, 조상클래스의 메서드를 자손클래스에서 오버라이딩한 경우에도 참조변수의 타입에 관계없이 항상 실제 인스턴스의 메서드(오버라이딩된 메서드)가 호출됨.





매개변수의 다형성

- 메서드의 매개변수에도 다형성이 적용됨.

- 부모클래스: Product 자식클래스 : Tv, Computer, Audio

그 외 클래스 : Buyer 클래스 있을 때,

* Buyer가 buy라는 메서드를 수행하면, 각각의 물건을 살때마다 buy 메서드를 구현하게 됨. Ex) void buy(Tv tv); void buy(Audio audio) …
* Product 타입의 참조변수를 매개변수로 받으면, 다형성이 적용되어

void buy(Product p) 로 자손타입의 참조변수면 매개변수로 받아들일 수 있게 됨.

PolyArgEx.java