

Casting(다형성)

부모타입의 참조변수로 자식타입의 객체를 다루는 것

하나의 객체가 여러가지 타입을 가질 수 있다.

캐스팅을 할 경우 사용할 수 있는 "멤버의 갯수"가 달라진다.

부모와 자식관계만 서로 형변환이 가능하다.

Monkey : 부모클래스(name, eat()), Human : 자식클래스(money, job, sleep())

```
Monkey mk = new Monkey();
```

// 상속관계일 경우 참조변수의 타입과 객체의 타입이 불일치 해도 ok

```
Monkey humon = new Human();
```

- 업 캐스팅(up Casting)

부모타입의 참조변수로 자식타입의 객체를 다루는 것(자식 >>> 부모)

필드의 사용범위가 부모타입 만큼 제한되기 때문에 자식 타입에서 추가된 내용은 사용할 수 없다.

```
부모클래스 참조변수 = new 자식생성자();
```

```
참조변수.부모필드
```

- 다운 캐스팅(down Casting)

업캐스팅 된 객체(부모타입의 참조변수)를 다시 자식타입으로 바꾸는 기법(부모 >> 자식)

다운 캐스팅 시 다시 자식 클래스의 필드를 사용할 수 있게 된다.

```
자식클래스 참조변수 = (자식클래스)업캐스팅 객체;
```

```
참조변수.자식필드
```

- 형변환 시 중요 point!

만든 인스턴스(객체)가 무엇인지가 중요하다!!

실제 인스턴스의 멤버갯수를 넘어서는 안된다.

- Casting을 사용 이유?

여러개의 자식객체를 한번에 관리 해야할 때가 있다.

여러 자식객체를 전달받기 위해서는 부모타입의 저장공간으로 받아야 한다.

그래서 up Casting을 통해 서로 다른 타입의 자식 객체들을 부모타입으로 만들어

하나의 객체(부모타입)에 모든 자식을 받을 수 있게 한다.

>> 그럼 부모클래스가 여러 자식클래스들을 관리할 수 있게 된다.

instanceof

객체 간 타입을 비교 해 참조변수의 형변환이 가능한지 확인

객체 instanceof 클래스명 : 객체가 뒤에오는 클래스의 타입인지 비교

만약 객체가 해당 클래스 타입이라면 true, 아니라면 false 반환

객체 샘플

