

타입 변환과 다형성

다형성

- 다형성은 사용방법은 동일하지만 **다양한 객체를 이용해서 다양한 실행결과**가 나오도록 하는 성질
(예. 타이어 교체, 무기 교체)
- 다형성을 구현: 메소드 재정의 + 타입변환 => 다형성

자동 타입 변환(업 캐스팅)

- 클래스의 변환은 **상속관계에 있는 클래스** 사이에 발생
- **자식 -> 부모** 타입으로 자동 타입 변환이 가능
- Animal 클래스와 Cat 클래스가 있으면 **“고양이는 동물이다”**가 성립하므로 자동 타입변환 가능

```
Cat cat = new Cat();  
Animal animal = cat;
```

```
Animal animal2 = new Cat();
```

자동 타입 변환 (업 캐스팅)

- 부모 타입으로 자동 타입 변환된 이후에는 **부모클래스에 선언된 멤버변수와 메소드만 접근 가능**
- 예외) 메소드가 자식클래스에 **재정의(Override)** 되어있다면 **자식클래스의 메소드가 대신 호출됨.**

```
5 class Animal {  
6     void methodA(){ System.out.println("부모"); }  
7 }  
8 class Cat extends Animal {  
9     @Override  
10    void methodA() { System.out.println("자식"); }  
11 }  
12  
13 public class Main {  
14  
15    public static void main(String[] args) {  
16  
17        Cat cat = new Cat();  
18        Animal animal = cat;  
19  
20        animal.methodA();  
21  
22    }  
23 }
```

Console
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (2021)
자식

멤버변수와 매개변수의 다형성

- 마치 자동차의 타이어를 교체 하듯
다른 객체로 교체(변경) 가능
- 멤버변수와 마찬가지로 매개변수에서도
자동 타입 변환이 일어난다

```
class Weapon {}
class Sword extends Weapon {}
class Gun extends Weapon {}
class Bow extends Weapon {}

class Hero {
    private Weapon w;

    void setWeapon(Weapon w) {
        this.w = w;
    }
}

public class Main {

    public static void main(String[] args) {
        Hero hero = new Hero();
        hero.setWeapon(new Sword());
        hero.setWeapon(new Gun());
        hero.setWeapon(new Bow());
    }
}
```


강제 타입 변환(다운 캐스팅)

- 부모 -> 자식 타입으로 변환 할 때
- 단, 자식 -> 부모 변환 후 다시 부모 -> 자식 타입으로 변환 할 때에만 강제 타입 변환 가능

```
Child c = (Child)p;
```

즉, 처음부터 부모타입으로 생성된 객체는 자식타입으로 변환 불가!

```
9 public class Main {
10
11     public static void main(String[] args) {
12
13         Parent p1 = new Parent();
14         Child c1 = (Child)p1;    // X
15
16         Parent p2 = new Child();
17         Child c2 = (Child)p2;    // O
18
19     }
20 }
21 }
```

Console

<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (2021. 12. 22. 오후 3:44:04)

Exception in thread "main" [java.lang.ClassCastException](#): Parent cannot be cast to Child
at Main.main(Main.java:14)

객체 타입 확인

- 객체가 어떤 클래스의 인스턴스인지 확인하기 위해 **instanceof** 연산자를 사용!
- `boolean result = 객체(인스턴스) instanceof 타입(클래스)`

```
9 public class Main {
10
11     public static void main(String[] args) {
12
13         Parent p1 = new Parent();
14         Parent p2 = new Child();
15
16         System.out.println("p1이(가) Child의 인스턴스인가? " + (p1 instanceof Child));
17         System.out.println("p2이(가) Child의 인스턴스인가? " + (p2 instanceof Child));
18     }
19 }
20
21
```

Console

<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (2021. 12. 22. 오후 3:56:34)

p1이(가) Child의 인스턴스인가? false

p2이(가) Child의 인스턴스인가? true