

10. 상속

1) 상속

특정 클래스가 다른 클래스로부터 데이터(필드=변수)와 메소드(함수=기능)를 상속을 통해서 기존에 만들어진 것을 활용하는 것이다. 이미 검증되고 기존의 활용성이 높은 프로그램을 사용한다는 것에서 유리하고 시간을 단축하고 완성도를 높이는등 합리적으로 활용할수있다.

- 두 클래스를 부모(super클래스, 상위 클래스)와 자식(하위 클래스)으로 관계를 맺어줄수있다.
- 자바에서는 하나의 클래스는 하나의 클래스만 상속할수있다.(다중클래스를 지원하지 않는다.)
- 상속받은 클래스(자식클래스)를 실행하면 상속받은 부모클래스의 생성자를 먼저 생성 후 자식클래스의 생성자 함수가 생성된다.
- 대체로 부모클래스에 매개변수가 없는 생성자 함수가 정의되어야 한다.
- final 이 명시된 클래스는 상속이 불가능하다.

* 모든 자바 클래스 파일의 최상위 클래스에는 Object라는 클래스가 존재한다. 기본적으로 제공되는

System.out.println(), toString(), Scanner 등등.. 직접적으로 상속하진 않지만 자바는 모든 클래스 파일을 Object 하위 클래스로 자동 설정 돼있다.

즉, 자바 라이브러리나 유저가 만든 모든 클래스는"Object" 클래스를 부모클래스로 상속 받아서 사용 하게 된다.

2) 다형성

다형성 하나의 객체가 여러가지 타입을 갖는 것, 하위클래스 객체가 상위 클래스의 자료형으로 변환 가능하다

하위 클래스가 상위 클래스의 정보를 함유하고있지만 상위 클래스는 하위 클래스의 정보를 포함하고 있지 않다

```
1 class A{
2     public String x(){return "A.x";}
3 }
4 class B extends A{
5     public String x(){return "B.x";}
6     public String y(){return "y";}
7 }
8 class B2 extends A{
9     public String x(){return "B2.x";}
10 }
11
12 class Main {
13     public static void main(String[] args) {
14         A obj = new B();
15         A obj2 = new B2();
16         System.out.println(obj.y()); //실행되지 않는다.
17         System.out.println(obj.x()); // "B.x"출력
18         System.out.println(obj2.x()); // "B2.x"출력
19         //클래스 B의 인스턴트인 obj 를 클래스A형으로 데이터타입을 주었을때
20         //클래스 A에 존재하는 멤버만이 클래스 B의 멤버가 된다.
21         //동시에 클래스 B에서 오버라이딩한 멤버의 동작방식은 그대로 유지된다.
```

3) instanceof

instanceof 연산자는 자기자신의 클래스 객체인지, 어떤 클래스를 상속받았는지를 확인하는 연산자이다.

- 결과값은 boolean값으로 리턴된다.
- 객체 instanceof 타입클래스 형식으로 실행된다.

Ex)

```
1 class A {}
2 class B extends A{}
3
4 class TestMain{
5     public static void main(String[] args) {
6         A a = new A();
7         B b = new B();
8
9         System.out.println(a instanceof A); //true
10        System.out.println(b instanceof A); //true
11        System.out.println(a instanceof B); //false
12        System.out.println(b instanceof B); //true
13    }
14 }
```

4)this. this() & super. super()

1. this. 명령어는 해당 객체가 갖고있는 변수나 메서드를 가르킨다.
2. this() 명령어는 내 객체 클래스의 생성자를 가르킨다.

괄호안의 매개변수 개수로 오버로딩된 생성자를 구분할수있다.

3. super 명령어는 해당 클래스의 상속받은 상위클래스(부모클래스)의 변수나 메서드를 가르킨다.
4. super() 명령어는 해당 클래스의 상속받은 상위클래스(부모클래스)의 생성자 함수를 가르킨다.

보통 상속받은 자식클래스의 생성자 함수에서 같은 매개변수를 가진 부모클래스 생성자 함수의 초기화에 이용되는데 이때 Super()문장은 반드시 자식 생성자 함수 내 첫번째 라인에 정의 해야한다.

5)overloading & overriding

1. overloading: 함수의 중복정의

- 같은 이름의 함수를 매개변수(인자==파라미터)를 다르게 하여 정의하는 것이다
- 같은 이름이지만 인자가 다른 여러 함수를 중복 정의 할수있다.

2. overriding: 함수 재정의

- 오버라이딩은 함수명과 받는 매개변수, 리턴타입도 똑같이 정의하되 해당 함수의 로직을 덧붙이거나 필요에 따라 재정의 하는 것을 말한다. 해당 함수는 하나만 존재할 수 있고 그 함수를 상속받아 재정의 하여 사용하는것이다.

- 오버라이딩의 접근제한자는 재정의 할때 제한하는 범위가 같거나 커야한다.

Ex)protected 메서드를 public 으로 오버라이딩 할 순 있어도 범위가 작은 (default)나 private로는 불가능하다

*즉 함수의 매개변수를 다르게 하여 여러 개를 정의할 수 있는 건 오버로딩이고

같은 이름의 함수를 상속받아 필요에 따라 로직을 다르게 정의하는 건 오버라이딩이라고 할 수 있다.

6) toString()

toString()문은 해당 객체(인스턴트)의 정보를 String형식으로 나타내준다.

Ex)

```
1 package com.dk.parctice_ex01_super;
2
3 public class TestMain {
4     public static void main(String[] args) {
5         SuperChild c2 = new SuperChild(10, 20);
6
7         System.out.println(c1.toString());
8         System.out.println(c1);
9
10        //com.dk.parctice_ex01_super.SuperChild@2f92e0f4
11        //com.dk.parctice_ex01_super.SuperChild@2f92e0f4
12
13        //c1객체의 패키지.클래스명@해시코드정보가 출력된다.
```

- 보통 .toString()문은 해당객체의 정보를 문자열로 나타내고자 할때 사용하는데 자바에서도 사용자가 재정의 (overriding)하여 사용할것을 권장하고 있다.
- 위 예제에서 c1.toString()이라고 명시하지 않고 c1이라는 객체명만 출력해도 같은 결과를 출력하는데 이는 "객체명"에 자동적으로 .toString()이 호출된다는 걸 알 수 있다.