7. 상속

1. 일상에서 알고 있는 상속과 비슷한 개념

A클래스가 B클래스부터 상속받을 수 있음 [데이터(속성)와 메소드(기능)을 상속받을 수 있음]

A객체는 B객체의 데이터 및 메소드를 이용할 수 있고 또는 변경할 수도 있음

-이미 잘 짜여진 프로그래밍을 상속받아 결과물의 퀄리티를 높일 수 있다.

-기존의 훌륭한 프로그램은 검증이 잘 되어 있어 버그도 없을 수 있다.

-다양한 객체 상속을 통해 하나의 객체로 묶을 수 있다.

2. 오버라이드(재정의)

데이터 타입이 다르면 같은 이름의 함수라도 다른 기능으로 중복정의가 가능하다

부모클래스의 함수를 자식클래스에서 재정의 하는 것.

자식클래스가 부모클래스를 상속하여 자식한테 없는 함수를 호출하면 부모클래스에 가서 해당 메소드를 찾게됨. 만약 부모클래스의 메소드를 자식클래스에서 동일한 이름으로 재정의하면 부모클래스의 메소드를 찾지 않고 자식클래스의 메소드를 호출하게되는 것

```
public ParentClass() {
    System.out.println("매개변수가 없는 생성함수 호출");
}
public ParentClass(int i) {
    System.out.println("매개변수가 있는 생성함수 호출");
}
public void method1() {
    System.out.println("ParentClass의 method1() 함수");
}
public void method2() {
    System.out.println("ParentClass의 method2() 함수");
```

```java
public class ChildClass extends ParentClass {

    public ChildClass() {
        System.out.println("매개변수 없는 ChildClass 생성함수 ");
    }
    public ChildClass(int i) {
        System.out.println("매개변수 있는 ChildClass 생성함수");
    }
    public void method3() {
        System.out.println("ChildClass의 method3() 함수");
    }
    @Override
    public void method1() {
        System.out.println("ChildClass의 method1() 함수");
    }
```

```java
public static void main(String[] args) {
    ParentClass p1 = new ParentClass();// 매개변수 없는 곳을 불렀기 때문에 이렇게 출력
    ParentClass p2 = new ParentClass(1);//매개변수 있는 곳을 불렀기 때문에 이렇게 출력
    p1.method1();//ParentClass에서 정의한 method1이 출력
    p1.method2();//ParentClass에서 정의한 method2가 출력
```

```java
    ChildClass c1 = new ChildClass(); // 부모단의 매개변수 없는 곳 거친뒤에 자식단의 매개변수 없는 곳 출력
    ChildClass c2 = new ChildClass(2); //부모단의 매개변수 없는 곳 거친뒤에 자식단의 매개변수 있는 곳 출력
    c1.method1(); //ChildClass에서 오버라이딩 했기 때문에 그 결과가 나오는 것
    c1.method2(); //ChildClass에서 오버라이딩 하지 않았기 때문에 부모단의 결과가 나오는 것
    c1.method3(); //ChildClass에서 메소드 정의를 했기 때문에 이 결과가 나오는 것
}
```