



JAVA

GU, DA HAE

상 속

생 성 자

protected

오버라이딩

instance of

abstract

interface

inner class

JAVA

객

체

상속

Inheritance

상속은 이미 정의되어 있는 **기존 클래스를 기반으로 새로운 클래스를 정의**하는 방법이다.

```
class Test extends 클래스명{  
    ...  
}
```

- 상속은 **부모에게 물려받는 것**을 의미한다.
- 새로 만든 클래스를 이미 정의된 클래스에서 상속받는다면 **멤버를 모두 물려받아 사용할 수 있다**.
- 부모가 되는 클래스를 **부모, 상위, 수퍼, 기반(base; 베이스) 클래스**라 하고, 자식이 되는 클래스를 **자식, 하위, 서브, 파생(derived; 유도) 클래스**라 한다.

J A V A

예제

```
class Upper{
    public int a;
    public void printUpper(){
        System.out.println("UPPER!!");
    }
}

class Lower extends Upper{
    public int b;
    public void printLower(){
        System.out.println("LOWER!!");
    }
}
```

```
public class Ex01{
    public static void main(String[] args){
        Lower lower = new Lower();
        lower.a = 10;
        lower.b = 20;
        lower.printUpper();
        lower.printLower();
    }
}
```

Lower 인스턴스

Upper 멤버

Lower 멤버

J A V A

상속, 생성자

Inheritance

하위 클래스의 생성자는 반드시 상위 클래스의 생성자를 호출한다.

하위 클래스에서 상위 클래스를 멤버 변수로 들고 있다고 생각해도 좋다.

하위 클래스 인스턴스

상위 클래스 멤버

하위 클래스 멤버

하위 클래스에서 상위 클래스의 생성자를 명시적으로 호출할 때는 키워드 `super`를 사용한다.

예제

```
class Upper{
    private int a;
    public Upper(int a){ this.a = a; }
    public void printUpper(){
        System.out.println("UPPER!!" + a);
    }
}

class Lower extends Upper{
    public int b;
    public Lower(int a, int b){
        super(10);
        this.b = b;
    }
    public void printLower(){
        System.out.println("LOWER!!" + b);
    }
}
```

```
public class Ex01{
    public static void main(String[] args){
        Lower lower = new Lower();
        lower.printUpper();
        lower.printLower();
    }
}
```

해보기

상속관계에 있는 Point, Circle 클래스의 생성자를 정의해 보자.

```
class Point{
    private int x;
    private int y;

    public void printPoint(){
        System.out.println(x + ", " + y);
    }
}

class Circle extends Point{
    private double radius;

    public void printCircle(){
        System.out.println("중심 좌표 : ");
        printPoint();
        System.out.println("반지름 : " + radius);
    }
}
```

해보기

상속 관계에 놓인 Car, HybridCar, HydrogenCar 클래스의 생성자를 정의하고 사용해보자.

```
class Car{ // 일반 자동차
    int gasoline;

    public void showGasoline(){ System.out.println("가솔린량 : " + gasoline); }
}

class HybridCar extends Car{ // 하이브리드 자동차
    int electric;

    public void showHybridCar(){
        showGasoline();
        System.out.println("전기량 : " + electric);
    }
}

class HydrogenCar extends HybridCar{ // 수소 자동차
    int hydrogen;

    public void showHydrogen(){
        showHybridCar();
        System.out.println("수소량 : " + hydrogen);
    }
}
```