

# Object & Class

## (1) Class & Object

```
class AirCon {  
    //변수 정의  
    String company;  
    String color;  
    int price;  
    int size;  
    int temp;  
  
    //메소드 정의  
    void powerOn(){  
        System.out.println("power on");  
    }  
    void powerOff(){  
        System.out.println("power off");  
    }  
    void tempUp(){  
          
    }  
    void tempDown(){  
        temp--;  
    }  
}
```

//객체의 변수와 메소드 사용 테스트

```
public class AirConUse {  
  
    public static void main(String[] args) {  
        //객체생성  
  
        //변수 사용  
        airCon.color = "White";  
        airCon.temp = 10;  
        airCon.price = 10000;  
  
        //메소드 사용  
        airCon.tempUp();  
        System.out.println(  
            airCon.powerOn();  
            airCon.powerOff();  
            airCon.tempDown();  
        );  
    }  
}
```

airCon.temp = 11, airCon.color = White, airCon.price = 10000원  
power on  
power off  
airCon.temp = 10, airCon.color = White, airCon.price = 10000원

//같은 클래스를 이용해서 여러 객체 만들기 테스트

```
public class AirConUse2 {
```

```
public static void main(String[] args) {
```

```
//변수 사용
```

```
airCon1.color = "White";  
airCon1.temp = 10;  
airCon1.price = 10000;
```

```
//메소드 사용
```

```
airCon1.tempUp();
```

```
System.out.println("airCon1.temp = " + airCon1.temp + ", airCon1.color = " + airCon1.color  
+ ", airCon1.price = " + airCon1.price + "원 ");
```

```
airCon1.powerOn();
```

```
airCon1.powerOff();
```

```
airCon1.tempDown();
```

```
System.out.println("airCon1.temp = " + airCon1.temp + ", airCon1.color = " + airCon1.color  
+ ", airCon1.price = " + airCon1.price + "원 ");
```

```
//두 번째 객체 생성
```

```
System.out.println("airCon2.temp = " + airCon2.temp + ", airCon2.color = " + airCon2.color  
+ ", airCon2.price = " + airCon2.price + "원 ");
```

```
//airCon2 변수에 airCon1 변수 참조 값 할당
```

```
airCon2 = airCon1;
```

```
System.out.println("airCon2.temp = " + airCon2.temp + ", airCon2.color = " + airCon2.color  
+ ", airCon2.price = " + airCon2.price + "원 ");
```

```
}
```

```
}
```

airCon1.temp = 11, airCon1.color = White, airCon1.price = 10000원

power on

power off

airCon1.temp = 10, airCon1.color = White, airCon1.price = 10000원

airCon2.temp = 0, airCon2.color = null, airCon2.price = 0원

airCon2.temp = 10, airCon2.color = White, airCon2.price = 10000원

## (2)Variable

```
//변수 종류 테스트
```

```
public class VariableKind {
```

```
//인스턴스 멤버 변수
```

```
int memVar;
```

```
//static 멤버 변수
```

```
static int staticVar;
```

```
public static void main(String[] args) {
```

```
//로컬 변수
```

```
int localVar;
```

```
}  
}
```

```
vk1.memVar = 0  
vk1.staticVar = 0  
VariableKind.staticVar = 0  
staticVar = 0  
vk2.staticVar = 20
```

### 3. Method

```
class Args{  
    int x;  
    void add(int x){  
        x = x + 50;  
    }  
    void add(Args arg){  
        arg.x = arg.x + 40;  
    }  
    void addNew(Args arg){  
        arg = new Args();  
    }  
    void add(int[] arr){  
        arr[0]++;  
    }  
}  
public class ArgsTest {  
    /**  
     * 인수 전달 방식 테스트  
     */  
    public static void main(String[] args) {  
  
        arg.x = 100;  
  
        arg.add(arg.x);  
        System.out.println("arg.x = " + arg.x);  
  
        arg.add(arg);  
        System.out.println("arg.x = " + arg.x);  
  
        arg.addNew(arg);  
        System.out.println("arg.x = " + arg.x);  
  
        arg.add(arr);  
        System.out.println("arr[0] = " + arr[0]);  
    }  
}  
W
```

```
arg.x = 100  
arg.x = 140  
arg.x = 140  
arr[0] = 1
```

```
public class VariableArgument {
```

//JDK 1.5에서 인자의 개수를 가변적으로 받을 수 있는 방식이 추가됨.

//가변적으로 넘어온 인자들은 배열 타입으로 받아진다.

```
void printInfo(String ...infos){  
    if(infos.length != 0){  
        for(int i=0;i<infos.length;i++){  
            System.out.println(infos[i]);  
        }  
    }  
    else{  
        System.out.println("인자가 없네요.");  
    }  
}  
public static void main(String[] args) {
```

```
    }  
}
```

인자 없이  
인자가 없네요.  
인자 하나  
홍길동  
인자 두 개  
홍길동  
직업능력개발  
인자 세 개  
홍길동  
직업능력개발  
1억