

인스턴스 멤버

## ■ 인스턴스 멤버

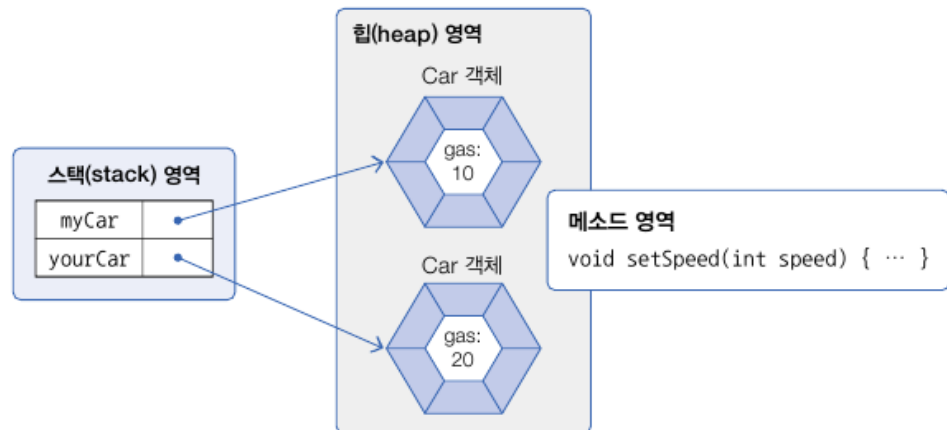
구분	설명
인스턴스(instance) 멤버	객체에 소속된 멤버 (객체를 생성해야만 사용할 수 있는 멤버)
정적(static) 멤버	클래스에 고정된 멤버 (객체 없이도 사용할 수 있는 멤버)

```
public class Car {
    //인스턴스 필드 선언
    int gas;

    //인스턴스 메소드 선언
    void setSpeed(int speed) { ... }
}
```

```
Car myCar = new Car();
myCar.gas = 10;
myCar.setSpeed(60);
```

```
Car yourCar = new Car();
yourCar.gas = 20;
yourCar.setSpeed(80);
```

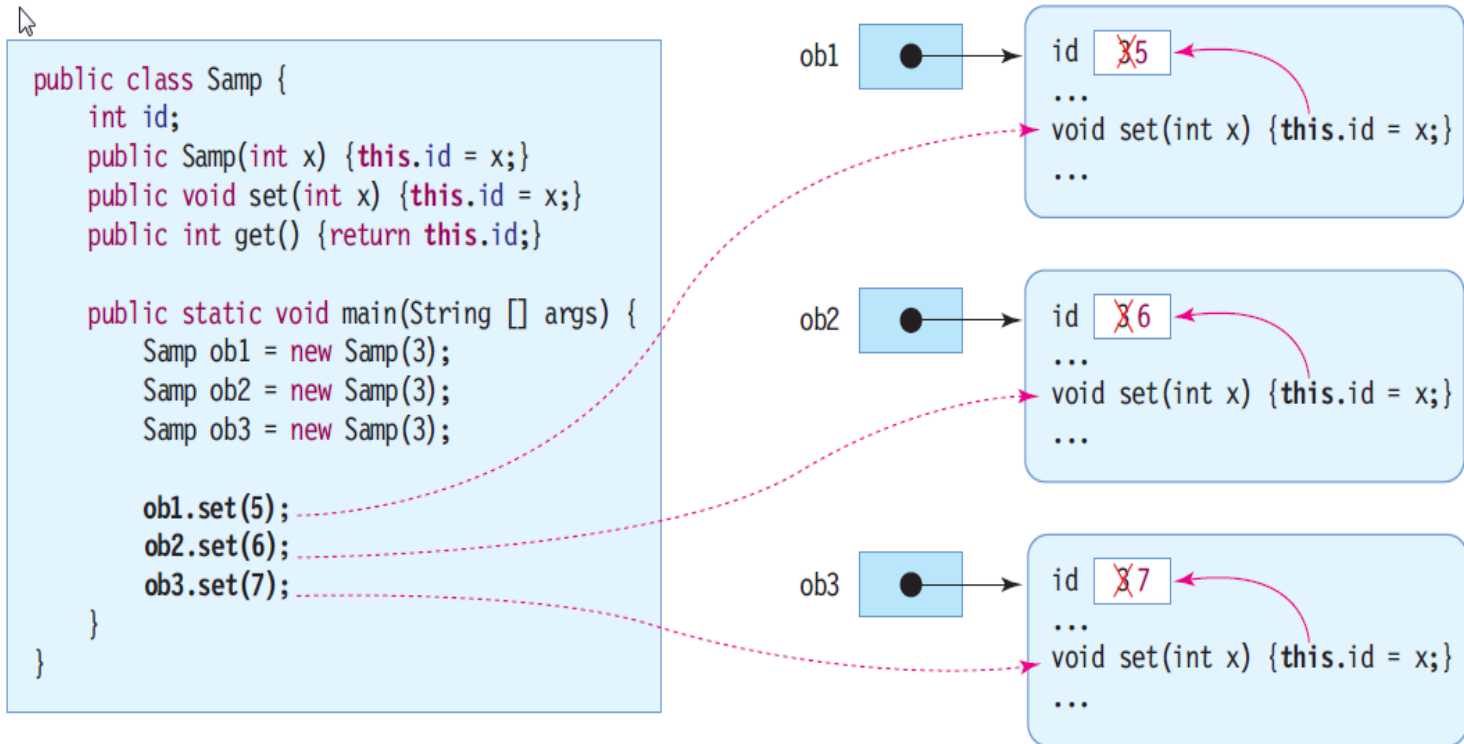


## ● this의 필요성

- 우리가 자신을 '나' 라고 하듯이, 객체는 자신을 'this'라고 한다.
- 객체 내부에서는 인스턴스 멤버에 접근하기 위해 this를 사용할 수 있다.
- 객체의 멤버 변수와 메소드 변수의 이름이 같은 경우
- 다른 메소드 호출 시 객체 자신의 레퍼런스를 전달할 때
- 메소드가 객체 자신의 레퍼런스를 반환할 때

```
class Samp {  
    int id;  
    public Samp(int id) { this.id = id; } // 매개 변수 이름과 필드의 이름이 같을 때  
    public void set(int id) { this.id = id; }  
    public int get() {return this.id; }  
  
    public Samp me() {  
        return this; // 자신의 레퍼런스를 반환할 때  
    }  
}
```

- this 이해



- 객체의 치환은 객체가 복사되는 것이 아니며, 객체의 주소가 복사된다.

```
public class Samp {  
    int id;  
    public Samp(int x) {this.id = x;}  
    public void set(int x) {this.id = x;}  
    public int get() {return this.id;}  
  
    public static void main(String [] args) {  
        Samp ob1 = new Samp(3);  
        Samp ob2 = new Samp(4);  
        Samp s;  
  
        s = ob2;  
        ob1 = ob2; // 객체의 치환  
        System.out.println("ob1.id="+ob1.id);  
        System.out.println("ob2.id="+ob2.id);  
    }  
}
```

