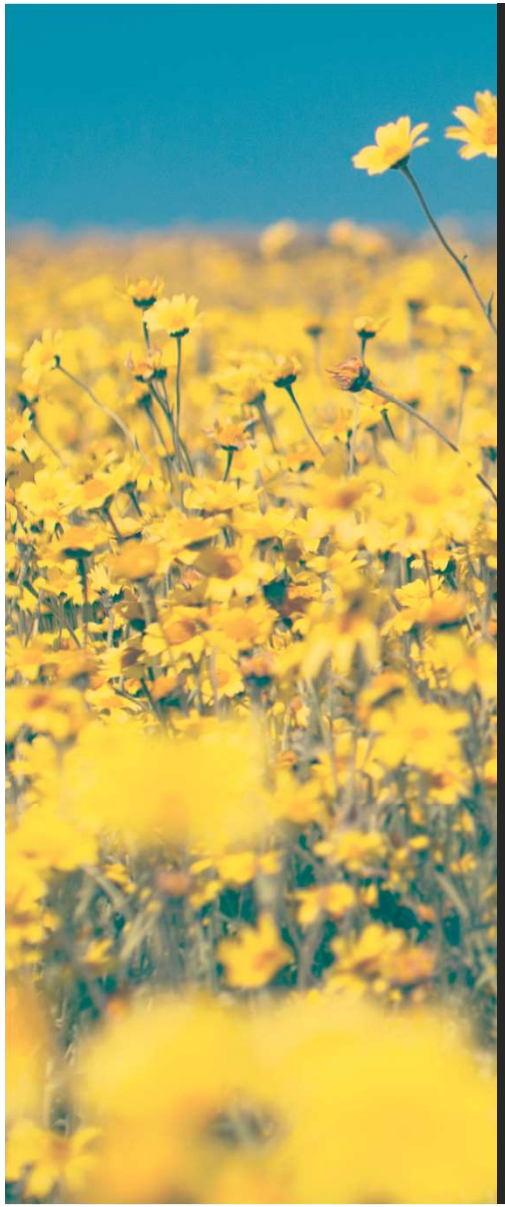


JAVA

GU, DA HAE


this
static 변수
static 메서드
JAVA
객체

JAVA

this

컴파일러가 모든 메서드에 자동으로 추가하는 매개변수. 메서드를 호출한 인스턴스의 위치를 기억한다.

```
class Test{
        ...
        void method(){
                ...
        }
        ...
}
```

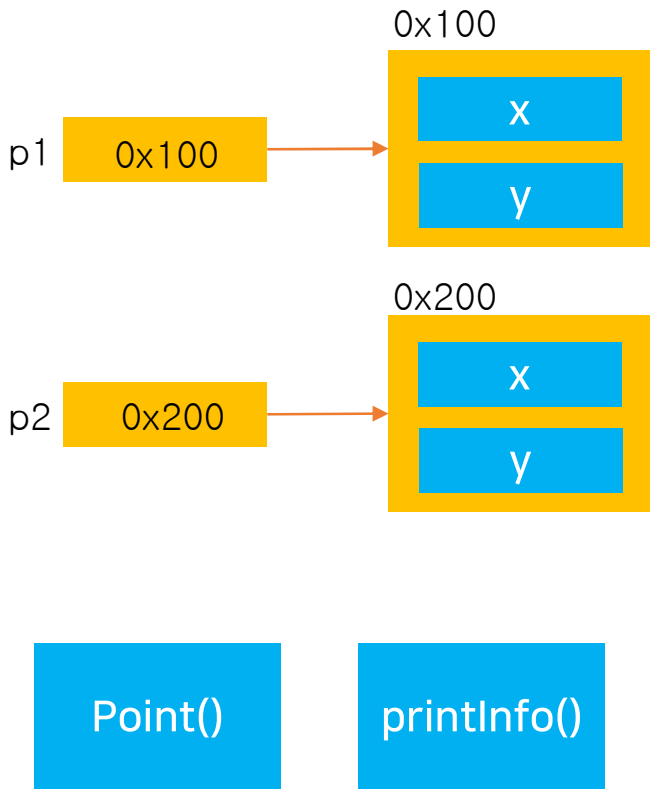
```
class Test{
        ...
        void method(Test this){
                ...
        }
        ...
}
```

메서드 내에서 인스턴스 멤버들을 참조할 때 사용하지만 생략하기 때문에 보이지 않는다.

JAVA

예제

```
class Point{
    int x, y;
    Point(int _x, int _y){
        x = _x;
        y = _y;
    }
    void printInfo(){
        System.out.println(x + ", " + y);
    }
}
public class Ex01{
    public static void main(String[] args){
        Point p1 = new Point(1, 2);
        Point p2 = new Point(200, 500);
        p1.printInfo();
        p2.printInfo();
    }
}
```



JAVA

예제

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

    Point(int _x, int _y){
        this.x = _x;
        this.y = _y;
    }
    void printInfo(){
        System.out.println(this.x + ", " +
        this.y);
    }
}
```

```
public class Ex01{
    public static void main(String[] args){
        Point p1 = new Point(1, 2);
        Point p2 = new Point(200, 500);

        p1.printInfo();
        p2.printInfo();
    }
}
```

JAVA

static 변수

모든 인스턴스가 공유하는 변수다. 클래스만 정의해도 접근이 가능하기 때문에 클래스 변수라고도 부른다.

```
static 자료형 식별자 = 값;
```

```
[접근 방법]

클래스명.식별자
인스턴스명.식별자
```

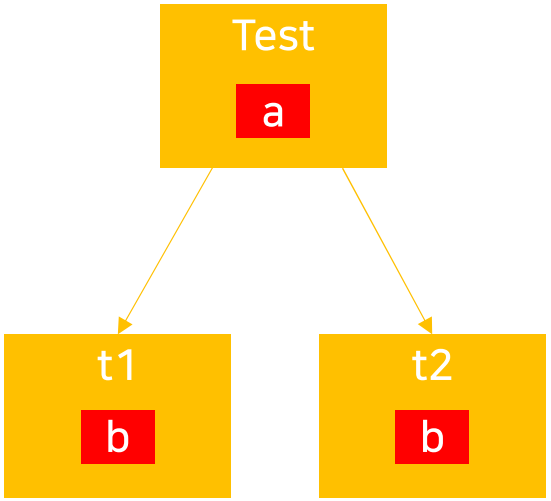
인스턴스들 간의 데이터를 공유할 때 사용한다.

JAVA

예제

```
class Test{
	static int a;
	int b;
}

public class Ex01{
	public static void main(String[] args){
		Test t1 = new Test();
		Test t2 = new Test();
		t1.a = 12345;
		System.out.println(t2.a);
	}
}
```



JAVA

예제

```
class Test{
    static int a = 0;
    int b;

    Test(){
        System.out.println(a +"번째 인스턴스 생성
        ");
        a++;
    }
}
```

```
public class Ex01{
    public static void main(String[] args){
        System.out.println(Test.a);

        Test t1 = new Test();
        Test t2 = new Test();

        t1.a = 12345;
        System.out.println(t2.a);
    }
}
```

JAVA

static 메서드

모든 인스턴스가 공유하는 메서드다. 클래스만 정의해도 접근이 가능하기 때문에 클래스 메서드라고도 부른다.

```
static 자료형 식별자(){
...
}
```

[접근 방법]

클래스명.식별자()
인스턴스명.식별자()

클래스를 정의하며 메서드만 멤버로 존재할 경우 static 메서드를 활용할 수 있다.

static 메서드는 매개변수 this가 없기 때문에 인스턴스 멤버에 접근할 수 없다.

JAVA

예제

```
class Test{

     static void m1(){
          System.out.println("m1()");
     }
     static void m2(){
          System.out.println("m2()");
     }
}
```

```
public class Ex01{
     public static void main(String[] args){
          Test.m1();
          Test.m2();

          Test t1 = new Test();
          Test t2 = new Test();
          t1.m1();
          t2.m1();

     }
}
```

JAVA

예제

```
class Calculate{ // 계산 클래스
    double add(double n1, doble n2){ return n1 + n2;}
    double min(double n1, doble n2){ return n1 - n2;}
    double mul(double n1, doble n2){ return n1 * n2;}
    double div(double n1, doble n2){ return n1 / n2;}

}
class Area{
    double rectangleArea(double w, double h){
        Calculate cal = new Calculate();
        return cal.mul(w, h);
    }
}
class Perimeter{
    double rectanglePerimeter(double w, double h){
        Calculate cal = new Calculate();
        return cal.add(cal.mul(2, w), cal.mul(2, h));
    }
}
```

```
public class Ex01{

    public static void main(String[] args){

        Area area = new Area();

        Perimeter peri = new Perimeter();


        System.out.println("직사각형 넓이 : " +
        area.rectangleArea(10, 20));

        System.out.println("직사각형 둘레 : " +
        peri.rectanglePerimeter(10, 20));


    }

}
```

JAVA

예제

```
class Calculate{ // 계산 클래스
	static double add(double n1, doble n2){ return n1 + n2;}
	static double min(double n1, doble n2){ return n1 - n2;}
	static double mul(double n1, doble n2){ return n1 * n2;}
	static double div(double n1, doble n2){ return n1 / n2;}
}

class Area{
	static double rectangleArea(double w, double h){
		return Calculate.mul(w, h);
	}
}
```

```
class Perimeter{
	static double rectanglePerimeter(double w, double h){
		return Calculate.add(Calculate.mul(2, w), Calculate.mul(2, h));
	}
}

public class Ex01{
	public static void main(String[] args){
		System.out.println("직사각형 넓이 : " + Area.rectangleArea(10, 20));
		System.out.println("직사각형 둘레 : " + Perimeter.rectanglePerimeter(10, 20));

	}
}
```

JAVA

예제

```
class Test{
	int n;

	static void setN(int _n){
		n = _n; // ERROR!!
	}
}

public class Ex01{
	public static void main(String[] args){
		Test.setN(20); // …??
	}
}
```

JAVA

예제

main 메서드의 위치

```
public class Point{
    int x;
    int y;
    Point(int x, int y){
        x = x;
        y = y;
    }
    void printInfo(){
        System.out.println(x + ", " + y);
    }
    public static void main(String[] args){
        Point p1 = new Point();
        p1.printInfo();
    }
}

// main 메서드는 어디에 만들어도 상관없다.
```

JAVA

해보기

학생을 표현한 Student 클래스를 정의해보자. 이름, 학년, 반 등의 멤버 변수가 있어야 하며 현재 학생이 몇 명 학원에 등록 되었는지 출력하는 기능, 학생의 정보를 출력하는 기능이 필요하다.

```
public static void main(String[] args){
      Student s1 = new Student("한혜윤", 1, 3);
      Student s2 = new Student("지서연", 3, 7);

      Student.showCount();

      Student s3 = new Student("김진수", 2, 5);

      Student.showCount();
      s1.printInfo();
      s3.printInfo();
}
```

```
[실행 결과]
등록 원생 수 : 2

등록 원생 수 : 3

이름 : 한혜윤 - 1학년 3반
이름 : 김진수 - 2학년 5반
```