



## *클래스 변수와 클래스 메소드*

## 클래스 변수와 클래스 메소드

- static 선언을 붙여서 선언하는 클래스 변수

```
public class InstCnt {  
    public static int instNum = 0;  
  
    public InstCnt() {  
        instNum++;  
        System.out.println("인스턴스 생성 : " + instNum);  
    }  
}
```

```
public class ClassVar {  
    public static void main(String[] args) {  
        InstCnt cnt1 = new InstCnt();  
        InstCnt cnt2 = new InstCnt();  
        InstCnt cnt3 = new InstCnt();  
    }  
}
```

- static으로 선언된 멤버 변수는 해당 클래스의 모든 인스턴스가 공유하는 변수
- 인스턴스 생성 시 새로 생성되는 것이 아니라 하나만 존재하며, 모든 인스턴스가 그 데이터 공유

## 클래스 변수와 클래스 메소드

### ● 클래스 변수의 접근 방법

```
public class AccessWay {  
    public static int num = 0;  
}
```

```
public class ClassAccess {  
    public static void main(String[] args) {  
        AccessWay way = new AccessWay();  
        way.num++;          // 인스턴스의 이름을 통한 접근  
        System.out.println(way.num);  
        AccessWay.num++;    // 클래스의 이름을 통한 접근  
        System.out.println(way.num);  
    }  
}
```

- 인스턴스의 이름을 통해 접근 가능
- 클래스의 이름을 통해 접근 가능

## 클래스 변수와 클래스 메소드

- 클래스 변수의 유용성

```
public class Circle {  
    public final static double PI = 3.1415;  
    private double radius;  
  
    public Circle(double radius) {  
        this.radius = radius;  
    }  
  
    void showPerimeter() {  
        double peri = (this.radius * 2) * this.PI;  
        System.out.println("둘레 : " + peri);  
    }  
  
    void showArea() {  
        double area = (this.radius * this.radius) * this.PI;  
        System.out.println("넓이 : " + area);  
    }  
}
```

## 클래스 변수와 클래스 메소드

```
public class CircleConstPI {  
    public static void main(String[] args) {  
        Circle c = new Circle(1.2);  
        c.showPerimeter();  
        c.showArea();  
    }  
}
```

- PI는 원주율로 결코 변하지 않는 수이므로 final 키워드로 상수로 선언
- PI는 모든 인스턴스에서 데이터를 가지고 있을 필요가 없으므로 클래스 변수로 선언
- 참조 목적만 가지고 있는 데이터는 클래스 변수로 만들어 사용

## 클래스 변수와 클래스 메소드

- 클래스 메소드 정의와 호출

```
public class NumberPrinter {  
    private int myNum;  
  
    // 클래스 메소드 (static 메소드)  
    public static void showInt(int n) {  
        System.out.println(n);  
    }  
    // 클래스 메소드  
    public static void showDouble(double n) {  
        System.out.println(n);  
    }  
  
    // 인스턴스 메소드  
    public void setMyNumber(int n) {  
        this.myNum = n;  
    }  
    // 인스턴스 메소드  
    public void showMyNumber() {  
        showInt(this.myNum);  
    }  
}
```

## 클래스 변수와 클래스 메소드

```
public class ClassMethod {  
    public static void main(String[] args) {  
        NumberPrinter.showInt(20); // 클래스 이름을 통한 클래스 메소드 호출  
        NumberPrinter np = new NumberPrinter();  
        np.showDouble(3.15); // 인스턴스 이름을 통한 클래스 메소드 호출  
        np.setMyNumber(75);  
        np.showMyNumber();  
    }  
}
```

- 클래스 변수에 접근하는 방법과 차이 없음

## 클래스 변수와 클래스 메소드

- main 메소드가 public이고 static인 이유는?
  - static : 정적(메모리에 저장)으로 변화
  - 자바 프로그램이 실행되면 제일 먼저 main 메소드를 찾아서 실행
  - 실행하여 main 메소드를 실행하려면 메모리 상에 main 메소드가 존재해야 함
  - 클래스 내에 존재하는 main 메소드의 호출이 이루어지는 영역은 클래스 외부이므로 public으로 선언



## 클래스 변수와 클래스 메소드

1. 다음 main 메소드와 함께 동작하는 Accumulator 클래스를 정의하시오.

```
public static void main(String[] args) {  
    for(int i = 0; i < 10; i++) {  
        Accumulator.add(i);  
    }  
    Accumulator.showResult();  
}
```

실행 결과로 "sum = 45"가 출력되어야 합니다.