

3부 클래스와 객체

- 14장. 클래스단위의 멤버를 위한 static

최문환



14장. 클래스단위의 멤버를 위한 static

1. 정적 멤버변수를 갖는 클래스
2. 정적 메서드
3. Math 클래스의 정적 멤버변수와 정적 메서드

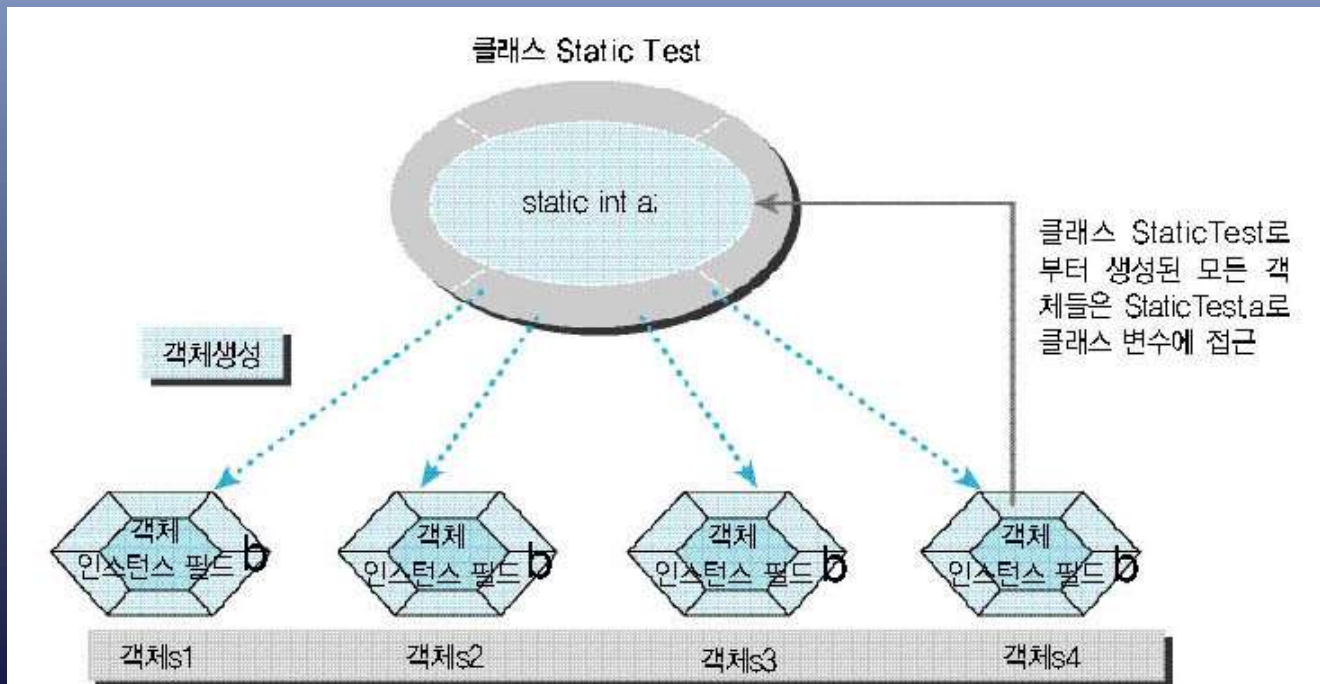
1. 정적 멤버변수를 갖는 클래스

모든 객체 인스턴스들이 하나의 멤버변수를 공유할 필요성이 있을 경우 `static`이란 예약어를 사용합니다.

1. 정적 멤버변수를 갖는 클래스

```
class StaticTest{  
    static int a=10;  
    int b=20;  
}
```

```
StaticTest s1 = new StaticTest();  
StaticTest s2 = new StaticTest();  
StaticTest s3 = new StaticTest();  
StaticTest s4 = new StaticTest();
```



<예제> 정적 멤버변수와 인스턴스 멤버변수의 차이점

```
001: class StaticTest{
002:   static int a=10;
003:   int b=20;
004: }
005: class StaticTest01 {
006:   public static void main(String[] args){
007:     System.out.println("StaticTest.a->" + StaticTest.a);
008:     StaticTest s1 = new StaticTest();
009:     StaticTest s2 = new StaticTest();
010:
011:     System.out.println("s1.a->" + s1.a + "\t s2.a->" + s2.a);
012:     System.out.println("s1.b->" + s1.b + "\t s2.b->" + s2.b);
013:
014:     s1.a=100;
015:     System.out.print("s1.a->" + s1.a );
016:     System.out.println("\t s2.a->" + s2.a);
017:
018:     s1.b=200;
019:     System.out.print("s1.b->" + s1.b);
020:     System.out.println("\t s2.b->" + s2.b);
021:   }
022: }
```

2. 정적 메서드

인스턴스 차원이 아닌 클래스 차원에서 사용하도록 설계하기 위한 정적메소드

```
public static int getA( ){  
    return a;  
}
```

클래스 차원에서 사용할 수 있는 메서드란 인스턴스의 생성 없이 클래스명으로 호출 가능하다는 의미

```
StaticTest.getA();
```

<예제> 정적 메서드 정의하기

```
001: class StaticTest{  
002:     private static int a=10;  
003:     private int b=20;  
004:  
005:     public static void setA(int new_a){  
006:         a = new_a;  
007:     }  
008:     public static int getA(){  
009:         return a;  
010:     }  
011: }
```

<예제> 정적 메서드 정의하기

```
012: public class StaticTest02 {  
013:     public static void main(String[] args) {  
014:  
015:         System.out.println(StaticTest.getA());  
016:  
017:         StaticTest s1=new StaticTest();  
018:         StaticTest s2=new StaticTest();  
019:  
020:         s1.setA(10000);  
021:         int res1=s1.getA();  
022:         System.out.println(res1);  
023:         System.out.println(s2.getA());  
024:     }  
025: }
```


2.2 정적 메서드와 인스턴스 메서드의 차이점

1. 정적 메서드에서는 this 래퍼런스를 사용할 수 없다.
2. 정적 메서드에서는 인스턴스 변수를 사용할 수 없다.

<예제>-정적 메서드에서 this 사용 불가능

```
001: class StaticTest{
002:   private static int a=10;
003:   private int b=20;
004:   public static void printA(){ //정적 메서드에서는 this를 사용하지 못함
005:     System.out.println(a);
006:     System.out.println(this.a); //컴파일 에러 발생
007:   }
008:
009:   public void printB(){ //this는 인스턴스 메서드에서 여러 객체에 의해서
010:     System.out.println(this.b); //메서드가 호출될 때 이를 구분하기 위해서 사용
011:   }
012: }
013: public class StaticTest03 {
014:   public static void main(String[] args) {
015:     StaticTest.printA();
015:     StaticTest s1 = new StaticTest();
016:     StaticTest s2 = new StaticTest();
017:     s1.printB();
018:     s2.printB();
019:   }
020: }
```

<예제>-정적 메서드에서 인스턴스 멤버 사용 불가능

```
001: class StaticTest{
002:   private static int a=10;
003:   private int b=20;
004:   public static void printA(){
005:     System.out.println(a);
006:     System.out.println(b); //컴파일 에러 발생
007:   }
008:
009:   public void printB(){
010:     System.out.println(b);
011:   }
012: }
013: public class StaticTest04 {
014:   public static void main(String[] args) {
015:     StaticTest.printA();
015:     StaticTest s1 = new StaticTest();
016:     StaticTest s2 = new StaticTest();
017:     s1.printB();
018:     s2.printB();
019:   }
020: }
```

<예제> Math 클래스의 정적 메서드의 사용 예

```
001: class StaticTest06 {  
002:     public static void main(String[] args) {  
003:         int a=40, b=30, c=10;  
004:  
005:         int res;  
006:         res=Math.max(a, b);  
007:         System.out.println(a + "와 " + b + " 중 최대값: "+res);  
008:  
009:         res=Math.max(b, c);  
010:         System.out.println(b + "와 " + c + " 중 최대값: "+res);  
011:     }  
012: }
```

<예제> Math 클래스의 정적 멤버변수의 사용 예

```
001: class StaticTest07 {  
002:     public static void main(String[] args) {  
003:         System.out.println(Math.PI);  
004:         int r=5;  
005:         double area;  
006:         area=r*r*Math.PI;  
007:         System.out.println("반지름이 "+r+"인 원의 면적 "+ area);  
008:     }  
009: }
```