Class Member - Class Variable

Class Member

- 옷만드는 디자이너의 줄자, 줄자는 여러 디자이너가 다 공유해서 쓴다.
- 줄자는 public하게 선언하면 외부에서 가져다 쓸 수도 있다.

Class Variable - Static Variale

- Instance와 아무 상관이 없는 변수
- Class 내에 선언하는 변수
- 인스턴스의 생성과 상관없이 메모리 공간에 할당되고, 초기화되는 변수
- public으로 선언하면 누구나 접근할 수 있다.

- JVM은 실행과정에서 필요한 클래스를 순간순간 Static 영역 메모리에 로딩하는데, 이 로딩 시점이 Static Variable이 메모리가 할당되는 시점이다.

```
class InstCnt {

     누구나 접근 가능한 변수인데 class 안에 자리를 빌려서 들어왔지만,
     계약관계에 의해 들어왔으므로 class의 일부는 아니다.

     static int instNum=0;

     public instCnt() {
          instNum++;
          System.out.println("인스턴스 생성 : "+instNum);
     }
}

class ClassVar {
     Public static void main(Stirng[] args) {
          InstCnt cnt1 = new InstCnt(); // instNum : 1
          InstCnt cnt2 = new InstCnt(); // instNum : 2
          InstCnt cnt3 = new InstCnt(); // instNum : 3
          // Static Variable instNum은 InstCnt Class 기반으로 생성된 모든 instance가 공유
     }
}
```

Access Static Variale

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

     AccessWay() { incrCnt(); }

     Public void incrCnt() { num++ }; // 클래스 내부에서 접근
}

class ClassVarAccess {
     public static void main(String[] args) {
          AccessWay way = new AccessWay();

          way.num++;

          /*
               instance를 통한 접근, 그러나 num이 instance Variable처럼 보일 수 있으므로 좋지 못한 표현.
```

```
                static variable num은 instance의 안에 존재하는 것이 아니라,
                instance가 참조변수 way에게 static 변수로 접근하라고 요청하는 것과 같다.
            */
            System.out.println("num = "+AccessWay.num); // Class 이름을 통한 접근
        }
}
```
