객체 : 속성과 기능을 갖춘것

클래스 : 속성과 기능을 정의한 것

인스턴스 : 속성과 기능을 가진것 중 실제 하는것

singeton pattern : 하나의 인스턴스만 생성하도록 구현 할 수 있음.

예제

package designpattern;

public class SingletoePattern {

public static void main(String[] args) {

// 기본 : 개발 중인 시스템에서 스피커에 접근 할 수 있는 클래스를 만들기 ==> 100개의 스피커 모두 볼륨오릴기

// 추가요청 : 인스턴스를 추가시마다 로그를 직음

//new 로 하면안됨

SystemSpeaker speaker1 = SystemSpeaker.getInstance();

SystemSpeaker speaker2 = SystemSpeaker.getInstance();

System.out.println(speaker1.getVolume());

System.out.println(speaker2.getVolume());

speaker1.setVolume(11);

System.out.println(speaker1.getVolume());

System.out.println(speaker2.getVolume());

speaker2.setVolume(22);

System.out.println(speaker1.getVolume());

System.out.println(speaker2.getVolume());

}

}

class SystemSpeaker{

static private SystemSpeaker instantce;

private int volume;

private SystemSpeaker(){

setVolume(5);

}

public static SystemSpeaker getInstance(){

if(instantce==null){

//시스템의 스피커에 접속

instantce= new SystemSpeaker();

//추가요청

System.out.println("새로생성");

}

//추가요청

System.out.println("이미생성");

return instantce;

}

public int getVolume() {

return volume;

}

public void setVolume(int volume) {

this.volume = volume;

}

}