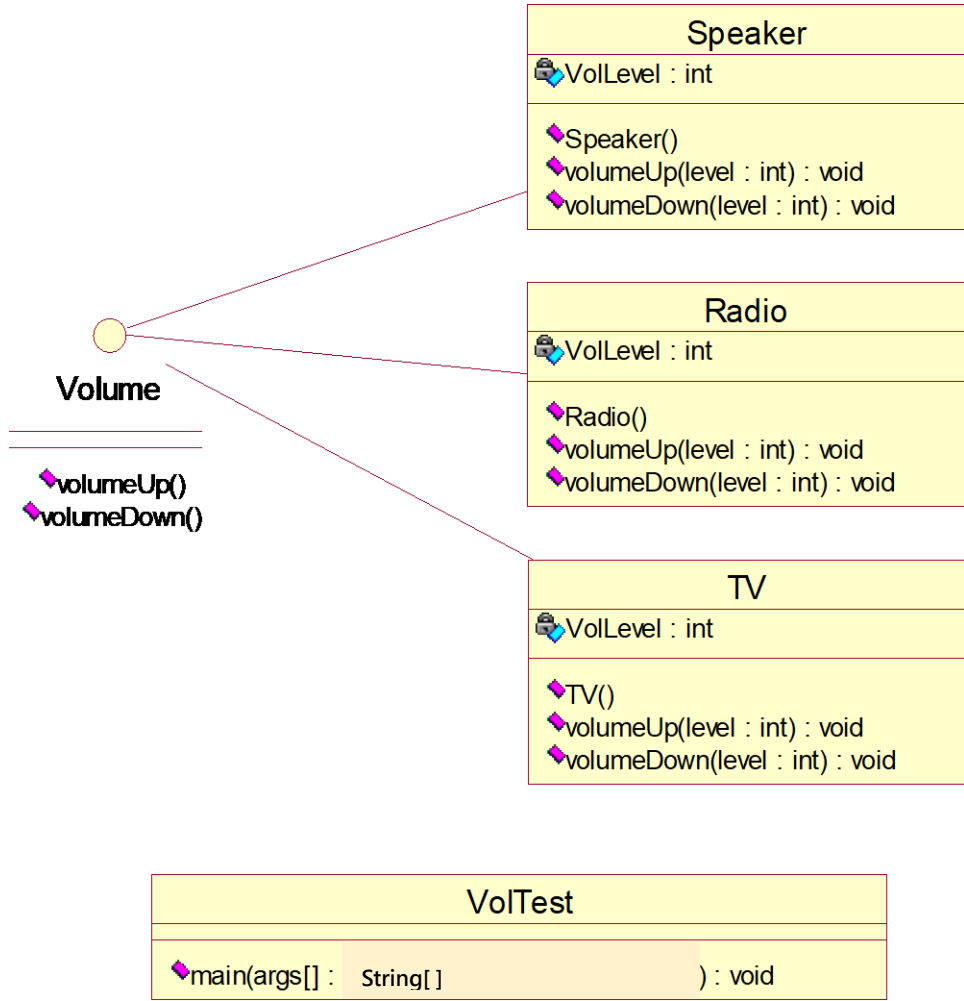


주어진 클래스 다이어그램을 프로그램 코드로 구현하는 문제입니다.

각 클래스에 상속되어진 메소드를 파일에 주어진 명제대로 재정의 하십시오.  
또한 VolTest 클래스에서는 Volume 의 배열을 생성하고 각 배열의 원소에는 TV 와 Speaker 와 Radio 클래스를 넣습니다. Volume 타입의 배열을 통해 서로 다른 객체의 동일한 메소드를 호출하게 됩니다.

참고) 객체 지향의 설계 원칙 중에서 “개방-폐쇄의 원칙(The Open-Closed Principle)” 이 있습니다. 인터페이스의 구현에는 개방되지만 인터페이스의 변경에는 폐쇄적이라는 원칙입니다. 이 원칙을 이용하여 인터페이스를 사용하는 외부에서는 일관된 참조를 할 수 있으며 인터페이스를 구현하는 내부에서는 다양한 구현부를 제공할 수 있다는 장점이 있습니다.



	volumeUp(int level)	volumeDown(int level)
Speaker	speakerLevel 을 주어진 level 만큼 증가, 메시지 출력, speakerLevel 의 최대값은 100 이므로 100 초과시 speakerLevel 은 100 으로 설정	speakerLevel 을 주어진 level 만큼 감소, 메시지 출력, speakerLevel의 최소값은 -100 이므로 -100 미만시 speakerLevel 은 -100 으로 설정
Radio	radioLevel 을 주어진 level 만큼 증가, 메시지 출력	radioLevel 을 주어진 level 만큼 감소, 메시지 출력
TV	tvLevel 을 주어진 level 만큼 증가, 메시지 출력	tvLevel 을 주어진 level 만큼 감소, 메시지 출력, tvLevel의 최소값은 0 이므로 tvLevel < 0 이면 0으로 설정

	main
VolTest	<p>인터페이스 Volume 타입으로 배열을 생성합니다.</p> <p>인터페이스에 세개의 객체를 할당하고 인터페이스 타입의 배열의 객체 개수 만큼 각 객체의 volumeUp(200) , volumeDown(300) 메소드를 호출합니다.</p> <pre> Volume v[ ] = new Volume[3]; v[0] = new Speaker(); v[1] = new TV(); v[2] = new Radio(); for(int i = 0; i &lt; v.length; i++){     v[i].volumeUp(200);     v[i].volumeDown(300); } </pre>