New( ) : 객체생성 키워드라고 부르고 객체 생성시 기존의 틀에 해당하는 class의 멤버 변수가 복사가 된다. 틀에 해당하는 class를 만든 뒤 틀을 이용해 메모리에 멤버변수를 가진 저장공간을 할당하는 방식

생성자 : 객체를 생성함과 동시에 멤버변수에 값도 넣어주는 편리한 얘가 생성자라는 method 이다.

public Tv() {

System.out.println("객체 생성시 내가 자동으로 호출된다.");

}

* Class 이름을 Tv로 했을 때 다음과 같이 이름이 동일하고 대문자로 시작하고 반환형이 없는 것이 생성자라고 하고 이후에 main에서 new를 이용해 객체를 생성하면 자동으로 위의 생성자가 실행된다.

public Tv(int c, int v, boolean o) {

ch = c;

volume = v;

onoff = o;

}

Tv tv1 = new Tv(100, 10, true);

* Tv라는 class를 만들 때 멤버변수에 값을 일일이 넣지 않고 한줄만에 끝내고 싶을 때 위와 같이 생성자를 이용하면 main에서 객체 생성하며 값을 넣어줌으로써 한줄로 생성과 값을 넣어줄 수 있다.
* 멤버 변수 값을 객체 생성시 자동으로 넣고 싶으면 생성자 method 사용한다. 클래스 이름과 동일하게 method 만들면 자동으로 호출된다.

public String toString() {

return ch + ", "+volume+", "+onoff+", ";

}

* 멤버 변수 값을 한번에 찍고 싶으면 toString이라는 method에 어떤 값을 반환할 것인지 입력되어 있는데 이 method를 변경하면 된다. to까지만 쓰고 crtl+스페이스 치면 자동으로 완성되며 안에 return에 멤버 변수들을 입력해주면 main에서 객체 복사한 것을 찍으면 내부의 값들이 전부 반환된다.

Class를 만들면 내가 만들지 않아도 입력 값 없는 생성자가 자동으로 생성되어 실행이 된다. 다만 설정된 기능이 없었기에 별다른 행동이 눈에 보이지 않았던 것이다.

public Phone(String c, int s) {

System.out.println("자동 호출");

color = c;

size = s;

}

public Phone() {

System.out.println("ㅐㅙㅙ");

}

* Overloading : method를 만들 때 하나의 이름으로 여러 개의 method를 만들 수 있는 특성이고 같은 이름의 method는 입력 값으로 구분을 한다.

public Phone(String color, ) {

System.out.println("자동 호출");

this.color = color;

}

* This. 라는 것이 멤버 변수임을 알려주는 것이고 이렇게 사용하면 같은 color 라고 사용할 수 있다.