05/30 월

정적 : 실행 전에 미리 고정되어있다. 언제나 있다.

동적 : 될 수도 안될 수도 있음.

객체 지향 : 확장, 수정에 용이

상태 → 변수 ; 변수는 변할 가능성이 있다.

가로 값 변경 → 언제나 부피에 영향

가로 바뀌면 부피도 같이 바뀌는 **동작**이 필요. 또는,

가로 세로 높이 한번에 부피 계산하는 동작. → 게으르다~ 필요할 때 계산하는 표현을 많이 쓴다.

동작 → 메소드;

----메인메소드에서 다른 클래스 안의 메소드를 불러와서 출력하는 것과

메인메소드에서 직접 나타내는 것과는 약간 다른 점이 있다.---

결과 자체는 같음.

일을 하는 주체가 다르다.

05\_27의 Q1Book.java 파일에서;

myBook.printAll();

→ System.out.println(myBook.title); 을 출력하면 값은 똑같음

But 동작의 주체가 다르기 때문에 객체 지향 기법에서는 의미가 약간 다르다.

Q3AboutMovie.java

이전엔 – 변수 값을 읽기 위해선 변수 값의 초기화가 필요, 컴파일 에러.

클래스를 불러오면 초기화 하지 않았어도 표기 → 문자열 null, 실수 0.0, 정수 0으로 표시됨.

필드 초기화를 하지 않아도, 다른 값으로 초기화 되고, 클래스 내부에 임시 값을 적어놓으면 그 값으로 초기화되어 출력된다.

Movie goodMovie = new Movie();

→ Movie() 의 값

**– (기본)생성자(constructor) -**

5\_30 // Box.java

기본생성자는 **기본값**으로 초기화 해준다.

// 정수형 -> 0

// 실수형 -> 0.0

// 논리형 -> false

// 참조형 -> null

// char -> 0

의 값으로 초기화.

// 생성자

public Movie(String t, double s, String d, int p) {

// 필드 초기화하는 문장들이 들어감. << 주 목적

// 초기화 하고 싶은 값의 타입을 파라미터에 적어줌.

}

Movie m = new Movie("새영화", 7.5, "새감독", 2022);

System.out.println(m.title);

System.out.println(m.score);

System.out.println(m.director);

System.out.println(m.publishYear);→ 전달한 값으로 필드를 초기화.

- this = 자기 자신을 참조;