-클래스(class) - > 형태를 나타내는 것  
 접근권한 class 클래스명 {  
 내용 클래스, 변수, 메소드만 추가로 쓸 수 있으며,  
 기본적으로는 생성자(constructor)를 만들 수 있음  
 }  
  
권장사항으로 클래스명은 대문자로 시작.  
클래스를 생성하였다고 바로 사용하는 것이 아님.  
클래스를 사용하려면 객체 생성이 필요함(실체화 단계 필요)

-객체 생성  
클래스명 ①변수명 ④= ②new ③생성자  
  
ex ) Test09s ts = new Test09s("Hello~!");  
;  
②이 객체가 생성되는 시점  
생성자의 경우 객체 생성시 1회만 실행됨(핸드폰 구매 후 초기 설정하는 행위 생각하면 됨)  
생성자가 변수에 실질적으로 담기는 것이 아니라 ‘주소’값이 담김.

-생성자 -> 객체 생성시 1회만 실행되는 것  
접근권한 클래스명(인자타입, 인자명 …..) { 인자는 0~n개(있을 수도 없을 수도 있음)  
 내용 \*인자 : 해당 기능이 호출될 때 받아야하는 값들  
  
}  
  
ex) public Test09s(String a) {  
 System.out.println("추가생성자 호출");  
 s = a;  
 }  
  
요즘에는 프로그래밍 중간에 생성자를 잘 사용하지 않음.  
생성자를 일회용 메소드로 생각하면 이해가 편함

-생성자 규칙  
1) 기본생성자(인자가 없는 것)은 생략이 가능하다.(자바 기본 기능에 의해)  
2) 만약, 추가 생성자가 존재할 경우 기본 생성자를 사용하려면 반드시 기본생성자를 선언해야 사용 가능하다.(추가 생성자 존재시 생략 기능이 사라짐)  
  
-메소드(method) - 동작, 기능  
접근권한 반환타입 메소드명(인자타입, 인자명, …) {  
 내용(반환타입이 void가 아니라면 return값;)  
  
관리 용이, 재사용성(코드 낭비 줄음)  
  
\*반환  
}  
반환타입(int, String, ……) : 결과물을 줄 때 사용. 반환할 것이 없을 때 void를 반환타입으로 함.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 값 | null | void |
| 공간 | 존재(O) | 존재(O) | 존재하지 않음(X) |
| 내용 | 존재(O) | 존재하지 않음(X) | 존재하지 않음(X) |

-변수 명명 규칙  
1) 단어의 첫글자만 대문자(saveButton) - 클래스랑 구분하기 위해 두 번째 단어부터 대문자 처리. 주로 개발에서 많이 사용.  
2) 단어사이에 \_를 추가하여 구분(save\_button - 디자이너 / SAVE\_BUTTON - DB)  
3) \_로 시작하고 단어 사이를 \_로 구분(\_save\_button - 옛날 개발 방식. 절차지향 프로그래밍에서 썼던 방식