클래스 작성법!  
1. 메서드  
 : 일반적인 함수의 특성을 가지고 있다!  
 1) 반환형 지원  
 2) 매개변수  
2. 메서드 중, 생성자  
 : 객체 생성 시, 초기화를 목적으로 무조건 호출되는 메서드  
 특징)  
 1) 클래스명과 반드시 일치해야 한다!  
 2) 생성자의 명시는 개발 시 생략이 가능하나, 만약 생략하면 에러가 발생하는 것이 아니라 컴파일러에 의해 대폴트 생성자가 관여된다! 이 때, 디폴트 생성자의 코드는 최소한의 관여만 하므로 로직이 없다! (결국 목적은 에러방지)  
 3) 반환형이 없다!!  
 : 만일 반환형을 두면? 에러가 발생하지 않는 것이 정말 무섭다! New 뒤에 오는 생성자로 생각하지 않기 때문에 객체 생성 시 호출되지 않는다!

3. 하나의 클래스내에 메서드명은 중복이 가능하다. (대부분의 유명한 객체지향 언어에서는 추세)  
Overloading(=메서드 중첩)  
  
왜? 메서드명은 소중하니깐, 명명규칙을 힘들게 고민했기 때문에 사소한 기능차이로 새로운 메서드명을 지정해야 한다는 것은 상당히 비효율적이다.

4. call by value (메서드 호출 시 값으로 호출하는 현상)  
: 변수의 값이 복사되어 전달되므로, 해당 변수는 영향을 전혀 받지 않는다.  
 call by reference (메서드 호출 시 그 값이 특히나 객체의 주소 값일 경우의 호출)  
: 변수의 값이 주소 값이므로, 해당객체가 영향을 받는다.

5. package and import!  
클래스들을 모아 놓은 집합을 package!  
왜 모아놓나? 클래스명의 중복 방지! 관련성 있는 클래스들끼리 묶어서 관리하려고!  
개발 시 환경 구축!  
: 프로젝트/ src - .java 패키지명 / Dog.java  
 bin - .class 패키지명 / Dog.class  
.java 를 .class로 만들되, 지정한 경로에 패키지를 생성까지 해주는 명령어!  
javac –d 생성경로 컴파일대상 .java!

특정 클래스를 패키지에 넣었을 때 외부의 클래스가 접근하기 위해서는 그 경로를 알아야 하는데, 이 때 사용하는 경로를 위한 환경변수는? Classpath