

01-23 [Java]

 소유자	 종수 김
 태그	

메서드(=함수)

특정 기능을 하는 함수

특징

- 함수에 값을 입력하면 함수가 가진 연산을 처리한 다음 결과를 출력
- 여러번 같은 계산을 해야 한다면 함수를 만들어두고 필요한 입력 값을 넣어서 동일한 결과 보장 및 재사용 가능
- 함수를 호출할 때는 외부에서 필요한 값만 입력하면 됨. (블랙박스처럼)

메서드 정의

선언 + 구현부 두가지로 나뉨

선언 : `public static int add(int a, int b)`

- `public` : 다른 클래스에서 호출할 수 있는 메서드 (접근제어자)
- `static` : 객체를 생성하지 않고 호출할 수 있는 정적 메서드
- `int` : 반환 타입 정의 (메서드의 실행 결과 반환 타입)
 - `void`처럼 반환 값이 없을 수 있음
 - 만약 반환 타입이 `void`가 아니라면 반드시 타입에 맞는 `return`이 필요함.
- `add` : 메서드 이름
- `(int a, int b)` : 메서드 호출 시 전달하는 입력 값 정의.
해당 변수들은 해당 메서드 안에서만 사용 (=parameter, =매개변수)

본문 : 메서드가 수행해야하는 코드 블록

- 메서드를 실행하는 곳에서는 메서드 네임과 역할은 알지만, 내용 본문은 몰라도 됨. (캡슐화)
- 반환 타입이 없는 경우 컴파일러는 마지막에 자동으로 `return;` 삽입
- 반환 타입이 존재하는 경우 메서드는 `return`을 만나면 바로 메서드를 종료

메서드 호출과 값 전달

- 자바는 항상 변수의 값을 복사해서 대입(* 매우 중요)

`int num1 = num2` 라고 해도, `num1` 변수에 `num2` 자체가 들어가는 것이 아님.

- primitive 타입의 변수는 값을 복사해서 사용 하므로, 원본에 절대 손을 대지 않음.
- reference 타입의 변수는 참조중인 메모리 주소를 복사해서 전달하는데, 전달한 메모리 주소에 어떤 특정 작업을 한다면 추 후 원본이 참조하는 메모리 주소에도 변경이 되어 있으므로 복사본이 원본에 영향을 줄 수 있음.

메서드와 형 변환

메서드를 호출할 때도 형 변환이 적용

큰(double) -> 작은(int)로 가므로 값 손실이 있을 수 있음.
작은(int) -> 큰(double) 로 가므로 자동 형 변환이 발생 (double)intValue; 인셈

메서드 오버로딩

동일한 메서드 이름, 매개변수가 다른 메서드 선언 방식

- 메서드 시그니처 = 메서드 이름 + 매개변수 타입(순서)
메서드를 구분할 수 있는 고유한 식별자나 서명
- 메서드 시그니처에 반환 타입은 존재하지 않으므로 반환 타입만 다른건 같은 메서드로 취급 때문에 오버로딩이 불가능