2023-03-10

Method

▼ 함수와 거의 동일하다.

- 함수: 단독 사용
- 메서드: 객체를 통해 사용

자주 사용되는 프로그램 코드를 정의하는 단위

main() Method : 자바 프로그램에서 수행의 시작점.

메서드생성방법

(1) 기능에 따라 메서드명 정하기(소문자로 시작. 단 생성자 메소드 제외!)
(2) 매개변수, 리턴값에 대한 사양을 정한다. (매개변수는 method overloading을 지원한다.)

오버로딩 :같은 공간(클래스)안에 동일한 이름의 메서드를 여러개 정의하는 것.

하지만 반드시 매개변수 사양은 달라야한다.

매개변수 : 메서드가 호출될 때 데이터를 전달받는 변수

아규먼트 : 메서드를 호출할 때 전달하는 데이터(식)

```
System.out.printf("%d %d %c %f", sum, sum * 10, 'A', Math.random());
[제어자] 리턴값타입 메서드명([매개변수 선언])
        수행문장;
        수행문장;
        [return 리턴값])

static void methodName(){
  // return;
}
static int methodName(){
  return;
}
static String methodName(){
  return;
}
```

메서드 사용방법

메서드 기능이 필요할 때 호출하기

```
//메서드 호출하기
메서드명([아규먼트]);
// 매기변수가 없다면 아규먼트를 주지 않지만, 매기변수가 있다면 반드시 아규먼트를 주어야한다.
변수 = 메서드명([아규먼트]); <- 리턴값이 있는 메서드에 한해서
```

기본형

```
// 값을 다루는 타입
byte(1)
short(2)
char(2)
int(4)
long(8)
float(6)
double(8)
boolean(1)
```

참조형

```
// 참조값을 저장하여 참조되는 대상을 다루는 타입 - 객체형
m1(int p1, int p2)
m2(int []p)
m3(int...p) --> m3(), m3(1,2,3), m3(10), m3(New int [] {10,20,30})
m4(char ch, int...p)
System.out.printf(String s, Object...o);
m5(int...p)
```

```
// Java 5 부터 가변아규먼트 구문 추가
// ...은 배열을 포함한다.
static int add(int... p) {
    int sum = 0;
    for (int i = 0; i < p.length; i++)
      sum += p[i];
    return sum;
  }
```