메소드

리턴 타입(return type)

- 리턴값이란 메소드를 실행한 후의 결과값
- 리턴타입은 결과값의 타입을 의미 (리턴타입이 없으면 void)
- 리턴타입이 있다고해서 **반드시 변수에 저장할 필요 없음**

```
class Calculator {

    double divide(int x, int y) { return (double)x/y; }
}

public class Main {
    public static void main(String args[]) {
        Calculator cal = new Calculator();
        cal.divide(10, 20);
    }
}
```

메소드 이름

- 식별자 규칙에 따라 작성
  - 관례적으로 소문자로작성 하되 뒤이어오는 단어의 첫 글자는 대문자
  - $,_ 를 제외한 특수문자 사용 X
  - 숫자로 시작 X

```
void run() {…}
void startEngine() {…}
String getName() {…}
int[] getScore() {…}
```

매개 변수

- 메소드가 실행할 때 필요한 데이터를 **외부로부터 전달받기 위함**
- 전달받은 매개변수로 메소드 구현로직에 사용
- 매개값의 타입과 매개변수의 타입이 달라도 **자동 타입변환이 일어날 수 있는 경우**에는 컴파일 에러가 발생하지 않음!

```
double divide(int x, int y) { return (double)x/y; }
```

```
byte b1 = 10;
byte b2 = 20;
double result = cal.divide(b1, b2);
```

리턴(return)문

- 리턴타입이 있는 메소드는 **반드시** 리턴(return)문을 사용해서 리턴값을 지정해야 함.
- 리턴(return)문 **이후의 실행문**은 컴파일 에러 발생!
- 리턴타입이 없는 void로 선언된 메소드에서 리턴(return)문은 **메소드 실행을 강제 종료**하는 용도로 사용됨.

메소드 호출

- **클래스 내부**의 다른 메소드에서 호출할 경우 **단순히 메소드 이름으로 호출**

```
메소드(매개값, … ); // 리턴타입이 없거나 받지 않을 경우
타입 변수 = 메소드(매개값, … ); // 리턴값이 있고, 받고 싶은 경우
```

- **클래스 외부**에서 호출할 경우에는 **객체 생성 뒤 도트(.)연산자를 이용해 호출**

```
클래스 참조변수 = new 클래스();
참조변수.메소드(매개값, …); // 리턴타입이 없거나 받지 않을 경우
타입 변수 = 참조변수.메소드(매개값,…); // 리턴값이 있고, 받고 싶은 경우
```