메서드

필요성

- 메서드를 이용하지 않은 예제
- 메서드를 이용한 예제

```
합(1~10) : 55
합(10~100) : 5005
합(100~1000) : 495550
```

메서드를 이용하면 얻을 수 있는 장점

- 중복 코드를 줄이고 코드를 재사용할 수 있다.
- 코드를 모듈화해 가독성을 높이므로 프로그램의 품질을 향상시킨다.

메서드를 이용하지 않은 예제

```
public static void main(String[] args) {
                int sum = 0;
                for (int i = 0; i <= 10; i++)
                        sum += i;
                System.out.println("합(1~10) : " + sum);

                sum = 0;
                for (int i = 10; i <= 100; i++)
                        sum += i;
                System.out.println("합(10~100) : " + sum);

                sum = 0;
                for (int i = 100; i <= 1000; i++)
                        sum += i;
                System.out.println("합(100~1000) : " + sum);
        }
```

메서드를 이용한 예제

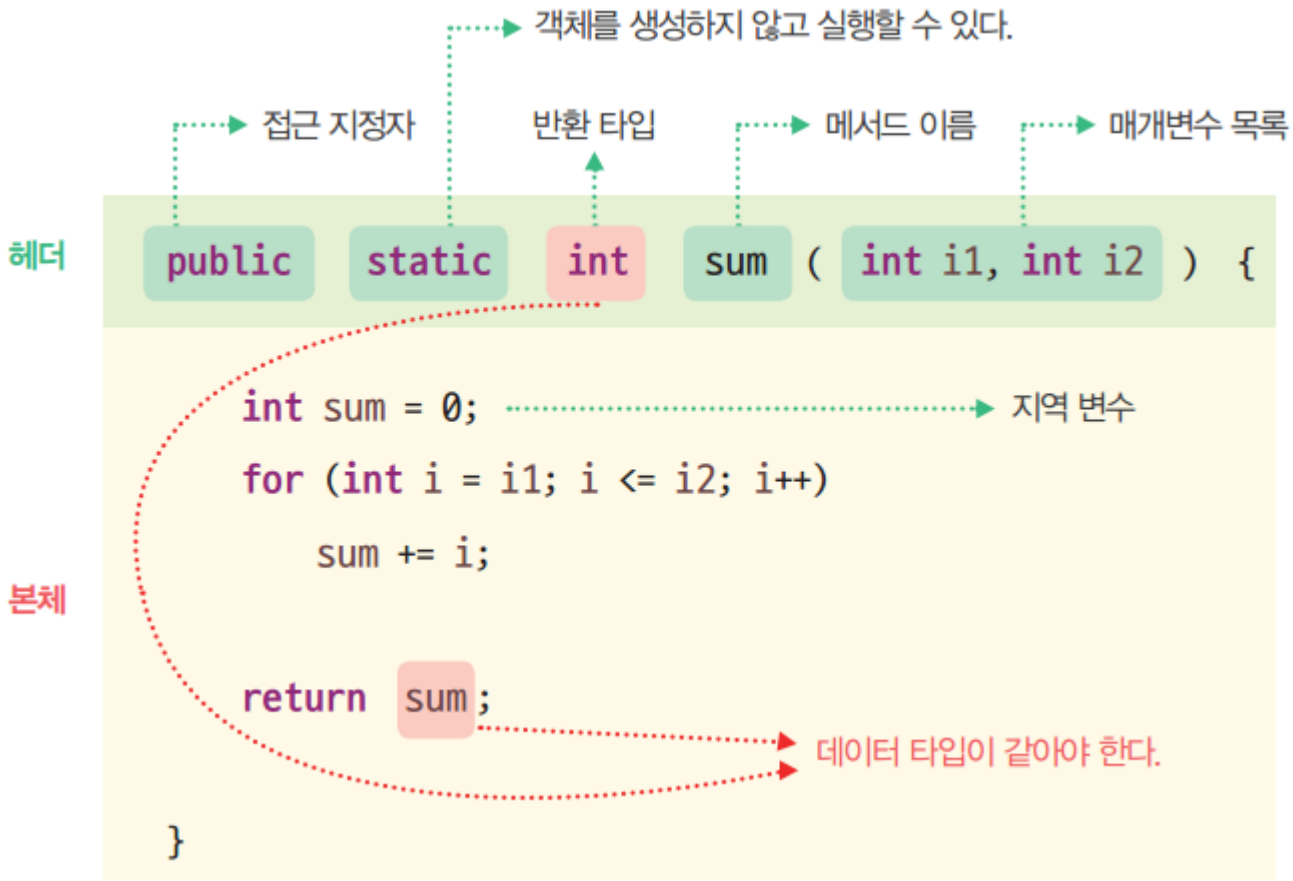
```
public static void main(String[] args) {
        System.out.println("합(1~10) : " + sum(1, 10));
        System.out.println("합(10~100) : " + sum(10, 100));
        System.out.println("합(100~1000) : " + sum(100, 1000));
}

public static int sum(int i1, int i2) {
        int sum = 0;
        for (int i = i1; i <= i2; i++)
                sum += i;

        return sum;
}
```

메서드

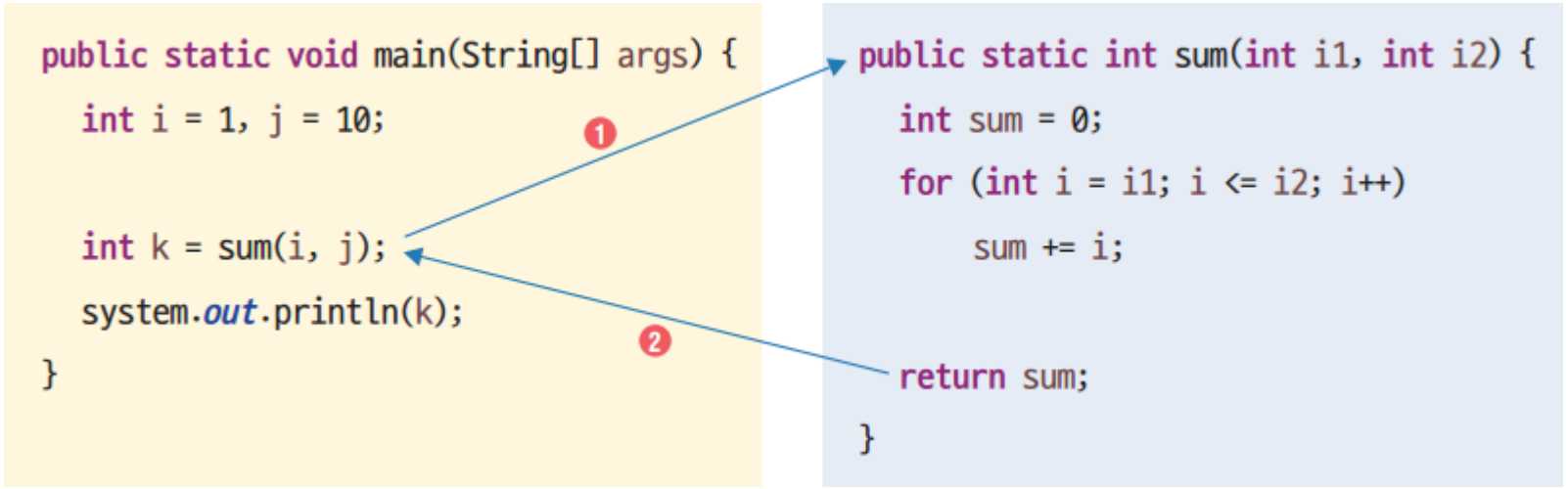
메서드의 구조


객체를 생성하지 않고 실행할 수 있다.
접근 지정자
반환 타입
메서드 이름
매개변수 목록
헤더
public static int sum (int i1, int i2) {
int sum = 0;
지역 변수
for (int i = i1; i <= i2; i++)
sum += i;
본체
return sum;
데이터 타입이 같아야 한다.
}

메서드

메서드의 호출과 반환

- 메서드를 호출하면 제어가 호출된 메서드(callee)로 넘어갔다가 호출된 메서드의 실행을 마친 후 호출한 메서드(caller)로 다시 돌아온다. 단, return 문을 사용하면 다음과 같이 메서드의 실행 도중에도 호출한 메서드로 제어를 넘길 수 있다.



- 예제

```
점수 : 99
잘못된 점수 : 120
```

리턴값 예제

```
public class ReturnDemo {
        public static void main(String[] args) {
                printScore(99);
                printScore(120);
        }

        public static void printScore(int score) {
                if (score < 0 || score > 100) {
                        System.out.println("잘못된 점수 : " + score);
                        return;
                }
                System.out.println("점수 : " + score);
        }
}
```

메서드

- **메서드의 매개변수**
  - 예제


```
public class EchoDemo {
        public static void main(String[] args) {
                echo("안녕!", 3);
        }

        public static void echo(String s, int n) {
                for (int i = 0; i < n; i++)
                        System.out.println(s);
        }
}
```

메서드

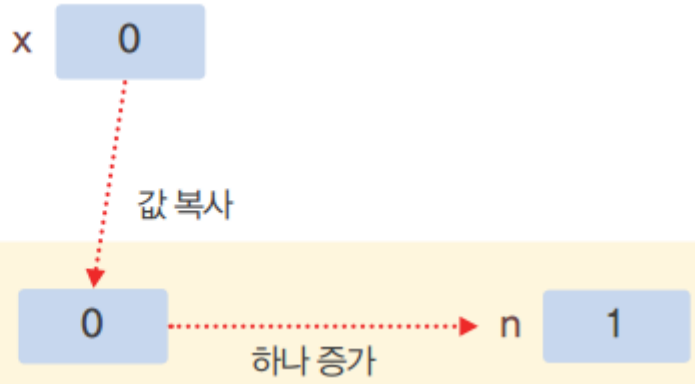
값 전달(call by value)

- 예제

```
increment() 메서드를 호출하기 전의 x는 0
increment() 메서드를 시작할 때의 n은 0
increment() 메서드가 끝날 때의 n은 1
increment() 메서드를 호출한 후의 x는 0
```

```
int x = 0;
increment(x);
// x는 여전히 0

increment(int n)
    n++;
```



값 전달 예제

```
public class IncrementDemo {
        public static void main(String[] args) {
                int x = 0;
                System.out.println("increment() 메서드를 호출하기 전의 x는 " + x);
                increment(x);
                System.out.println("increment() 메서드를 호출한 후의 x는 " + x);
        }

        public static void increment(int n) {
                System.out.println("increment() 메서드를 시작할 때의 n은 " + n);
                n++;
                System.out.println("increment() 메서드가 끝날 때의 n은 " + n);
        }
}
```