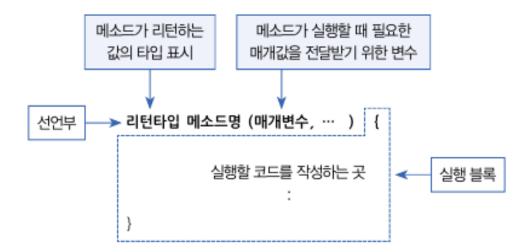
메소드 선언과 호출

■ 메소드 선언과 호출

◎ 메소드 선언



◎ 메소드 호출

```
객체 내부
             외부 객체
void method() {
                                             //생성자
 Calendar calc = new Calendar();
                                             Calculator() {
 calc.powerOn();
                                               powerOff(); -
 int r1 = calc.plus(3, 5);
 double r2 = calc.divide(15, 3);
                                                                                호출
                                             //메소드
                                       호출
                                            void powerOn() { ··· }
                                             void powerOff() { ··· }
              도트 연산자
                                             int plus(int x, int y) { ··· }
                                             double divide(int x, int y) { ··· }
                                                                                호출
                                             void method() {
                                               powerOn();
                                               int r1 = plus(3, 5);
                                               double r2 = divide(15, 3);
```

◎ 가변길이 매개변수

- 메소드가 가변길이 매개변수를 가지고 있다면, 매개변수의 개수와 상관없이 매개값을 줄 수 있다.

```
int sum(int ··· values) {
}
```

```
int result = sum(1, 2, 3);
int result = sum(1, 2, 3, 4, 5);

int[] values = { 1, 2, 3 };
int result = sum(values);

int result = sum(new int[] { 1, 2, 3 });
```

◎ return 문

- 메소드의 실행을 강제 종료하고 호출한 곳으로 돌아간다.
- 메소드 선언에 리턴 타입이 있을 경우에는 return 문 뒤에 리턴값을 추가로 지정해야 한다.
- return 문 이후에 실행문을 작성하면 컴파일 에러가 발생한다. 왜냐하면 return 문 이후의 실행문은 결코 실행되지 않기 때문이다.

```
int plus(int x, int y) {
  int result = x + y;
  return result;

System.out.println(result); //Unreachable code
}
```

◎ 메소드 오버로딩(overloading)

- 메소드 이름은 같되 매개변수의 타입, 개수, 순서가 다른 메소드를 여러개 선언하는 것을 말한다.

```
// 메소드 오버로딩이 성공한 사례

class MethodOverloading {
  public int getSum(int i, int j) {
    return i + j;
  }
  public int getSum(int i, int j, int k) {
    return i + j + k;
  }
  public double getSum(double i, double j) {
    return i + j;
  }
}
```

```
// 메소드 오버로딩이 실패한 사례

class MethodOverloadingFail {
  public int getSum(int i, int j) {
    return i + j;
  }
  public double getSum(int i, int j) {
    return (double)(i + j);
  }
}
```