

'메소드 오버로딩(Method Overloading)'은 클래스 내에 같은 이름의 메소드를 2개 이상 정의할 수 있게 해주는 기능입니다.

```
public class Calculator {
    int add(int a, int b) {
        return a + b;
    }

    int add(int a, int b, int c) {
        return a + b + c;
    }

    double add(double a, double b) {
        return a + b;
    }

    public static void main(String[] args) {
        Calculator c = new Calculator();

        System.out.println(c.add(2, 4));           // test 1
        System.out.println(c.add(2, 4, 8));        // test 2
        System.out.println(c.add(3.14, 2.54));     // test 3
    }
}

6
14
5.68
```

**add** 라는 이름의 메소드가 세 개나 있죠? 그런데 아무 오류도 나지 않습니다. 왜 그럴까요?

잘 보시면 각 메소드는 다른 구성의 파라미터들을 받습니다. 첫 번째 **add** 는 정수형 값 두 개, 두 번째 **add** 는 정수형 값 세 개, 그리고 세 번째 **add** 는 소수형 값 두 개를 받습니다.

같은 이름의 메소드라도 파라미터 구성이 다르면 자바는 알아서 구분을 할 수 있습니다. **main** 메소드에서 **c.add(2, 4)** 를 하면 알아서 첫 번째 **add** 가 불리고, **c.add(2, 4, 8)** 을 하면 두 번째 **add** 가 불리고, **c.add(3.14, 2.54)** 를 하면 세 번째 **add** 가 불리는 거죠.

## 쓰임 (Usage)

### 은행 계좌 예제

'달러'나 '유로'같은 다른 통화로 입금을 하는 **depositWithExchangeRate** 메소드를 아래처럼 쓸 수 있습니다.

```

public boolean deposit(int amount) {
    if (amount < 0 || amount > owner.getCashAmount()) {
        System.out.println("입금 실패입니다. 잔고: " + balance + "원, 현금: " + owner.getCashAmount() + "원");
        return false;
    }

    balance += amount;
    owner.setCashAmount(owner.getCashAmount() - amount);

    System.out.println(amount + "원 입금하였습니다. 잔고: " + balance + "원, 현금: " + owner.getCashAmount() + "원");
    return true;
}

public boolean depositWithExchangeRate(double amount, double exchangeRate) {
    return deposit((int) (amount * exchangeRate));
}

```

하지만 사실 **depositWithExchangeRate** 을 **deposit** 으로 바꿔도 아무 문제 없습니다. 파라미터 구성이 달라 '메소드 오버로딩' 개념이 적용되기 때문이죠!

```

public boolean deposit(double amount, double exchangeRate) {
    return deposit((int) (amount * exchangeRate));
}

```

## 콘솔 출력

생각해보면 지금까지 써왔던 **System.out.println** 도 파라미터를 굉장히 다양하게 받죠? 이게 다 '메소드 오버로딩'입니다.

```

System.out.println(5);           // 정수 파라미터
System.out.println(3.14);       // 소수 파라미터
System.out.println("hello");    // 문자열 파라미터
System.out.println(true);       // 불린 파라미터

```


```


5
3.14
hello
true

```



수업을 완료하셨으면 체크해주세요.

 수강생 Q&A 보기

 [\(/questions?](#)  
질문하기

assignment\_id=347&sort\_by=popular)  
[\(/questions/new?](#)

assignment\_id=347&op1=%EA%B0%9D%EC%B2%B4+%EC%A7

< 이전 강의  
메소드 오버로딩 (/assignments/336)

다음 강의  
계좌 이체 > (/assignments/338)