

15. 메서드 재정의(overriding)

오버라이딩은 상속 관계에 있는 클래스들 사이에 같은 이름의 메서드를 재정의하는 것이다. 오버라이딩은 부모클래스와 자식클래스에 있는 메서드의 이름이 같은 경우이다. 오버라이딩에서는 메서드 이름, 매개변수 개수 및 자료형, 반환형이 같아야 한다.

```

class A
{
    int compute(int a, int b){ return a + b; }
}
class B extends A
{
    int compute(int a, int b){ return a * b; } //메서드 재정의
}
class C extends B
{
    int compute(int a, int b){ return a - b; } //메서드 재정의
}
class OverrideDemo
{
    public static void main(String args[]){
        A ride1 = new A();
        B ride2 = new B();
        C ride3 = new C();

        System.out.println(ride1.compute(2, 3));
        System.out.println(ride2.compute(2, 3));
        System.out.println(ride3.compute(2, 3));
    }
}

```

[실행 결과]

5
6
-1

[Tip] OOP 언어에서 오버로딩과 오버라이딩(다형성 구현)

- Overloading : 동일한 클래스 내에서 같은 이름의 메서드를 정의하여 다형성 지원
- Overriding : 상속 관계의 클래스에서 같은 이름의 메서드를 정의하여 다형성 지원

//탐구 - 자바에서 오버로딩이냐? 오버라이딩이냐?

```
class A{
    int p = 1;
    int q = 2;
    int abc(int k){ return k + 1; }
}
class B extends A{
    int x = 100;
    int y = 200;
    int abc(int r, int s){ //여기서, 메서드 abc()는 별도로 취급됨(재정의된 것이 아님)
        return r + s;
    }
}
public class Test{
    public static void main(String args[]){
        A a = new A();
        B b = new B();

        a = (A)b; //상속 관계에 있을 때만 가능    [실행결과]
        System.out.println(a.p);                // 1
        System.out.println(a.q);                // 2
        System.out.println(b.p);                // 1
        System.out.println(b.q);                // 2
        System.out.println(b.x);                // 100
        System.out.println(b.y);                // 200
        System.out.println(a.abc(1000));        // 1001
        System.out.println(b.abc(2000));        // 2001 → 상속받은 것이 처리됨
        System.out.println(b.abc(3000, 4000)); // 7000    (재정의 아님)
    }
}
```

◆ 자바에서 메서드 재정의(Overriding)

- ① 메서드 이름이 같아도 인자목록과 반환형이 일치하지 않으면 재정의로 인정되지 않는다.
→ 새로운 메서드로 취급할 뿐이다.
- ② 메서드 이름과 인자목록은 같고 반환형이 다르면
두 메서드를 구별할 수 없어서 컴파일 오류가 발생된다.
- ③ 결론은 "메서드 이름, 인자목록, 반환형"이 같아야 재정의로 인정된다.

[Tip] 만약, 클래스 B에 다음 형태의 메서드가 있으면 재정의가 아니다.(오류 발생)

```
long abc(int k){ return k + 3; } → 반환형이 다르므로 오류 발생
```

기출문제 분석

1. 객체지향언어에서 클래스 A와 클래스 B는 상속관계에 있다. A는 부모클래스, B는 자식클래스라고 할 때 클래스 A에서 정의된 메서드(method)와 원형이 동일한 메서드를 클래스 B에서 기능을 추가하거나 변경하여 다시 정의하는 것을 무엇이라고 하는가? [2014년 국가 9급]

- ① 추상클래스(abstract class)
- ② 인터페이스(interface)
- ③ 오버로딩(overloading)
- ④ 오버라이딩(overriding)

♣ 오버라이딩(overriding, 재정의)

```

class A
{
    int don = 100;
    int get() { return don + 1; }
}

class B extends A           //상속
{
    int don = 200;
    int get() { return don + 2;} //재정의 - 다형성 지원 요인
}

class Test
{
    public static void main(String args[])
    {
        A p = new B();           //명령어 new가 객체(p)를 생성한다.
        System.out.println(p.don); //100 출력 - 속성 호출
        System.out.println(p.get()); //202 출력 - 재정의된 메서드 호출
    }
}

```

정답 : ④

2. 객체지향 프로그래밍의 특징 중 상속 관계에서 상위클래스에 정의된 메서드(method) 호출에 대해 각 하위클래스가 가지고 있는 고유한 방법으로 응답할 수 있도록 유연성을 제공하는 것은?
[2015년 국가 9급]

- | | |
|---------------------|----------------------|
| ① 재사용성(reusability) | ② 추상화(abstraction) |
| ③ 다형성(polymorphism) | ④ 캡슐화(encapsulation) |

↳ 객체지향에서 다형성

-
- 다형성은 오버로딩(overloading)이나 오버라이딩(overriding)으로 구현할 수 있다.
 - Overloading : 동일한 클래스 내에서 같은 이름의 메서드를 정의하여 다형성 지원
 - Overriding : 상속 관계의 클래스에서 같은 이름의 메서드를 정의하여 다형성 지원
-

정답 : ③