

# 자바 클래스 상속 및 오버로딩 연습

Sangkon Han  
27 October 2025

## 1 연습 문제: 이커머스 할인 정책 계산기 만들기

처음에는 모든 할인 로직이 하나의 메서드에 복잡하게 얹혀있다고 가정합니다.

### 1.1 기존 코드 기반으로 클래스 및 메서드로 분류

먼저, 할인 유형( discountType )에 따라 분기 처리가 되는 지저분한 원본 코드가 있습니다. 이 코드를 RateDiscountPolicy (정률 할인)와 AmountDiscountPolicy (정액 할인) 두 개의 개별 클래스로 분리해 보세요. 각 클래스는 calculateDiscount(int price) 메서드를 가져야 합니다<sup>1</sup>.

```
public class Main {
    public static void main(String[] args) {
        int priceA = 10000;
        String policyA = "RATE"; // 10% 할인
        int policyA_Value = 10;

        int priceB = 20000;
        String policyB = "AMOUNT"; // 1500원 할인
        int policyB_Value = 1500;

        int finalPriceA = 0;
        int finalPriceB = 0;

        // A 상품 할인 계산
        if (policyA.equals("RATE")) {
            finalPriceA = priceA - (priceA * policyA_Value / 100);
            System.out.println("A 상품 최종 가격 (정률 할인): " +
finalPriceA);
        } else if (policyA.equals("AMOUNT")) {
            finalPriceA = priceA - policyA_Value;
            System.out.println("A 상품 최종 가격 (정액 할인): " +
finalPriceA);
        }

        // B 상품 할인 계산 (로직 중복)
        if (policyB.equals("RATE")) {
            finalPriceB = priceB - (priceB * policyB_Value / 100);
            System.out.println("B 상품 최종 가격 (정률 할인): " +
finalPriceB);
        } else if (policyB.equals("AMOUNT")) {
            finalPriceB = priceB - policyB_Value;
            System.out.println("B 상품 최종 가격 (정액 할인): " +
finalPriceB);
        }
    }
}
```

<sup>1</sup>아래 코드를 읽고, 직접 코드를 작성해 보세요. 실행 후 이상이 있거나 개선사항이 있으면 코드를 직접 수정하시고, 문제가 없다면 계속 진행하세요

### 1.2 상속을 통해서 추상화

1단계에서 만든 두 클래스( RateDiscountPolicy , AmountDiscountPolicy )는 ‘할인 금액을 계산한다’는 공통의 목적을 가집니다.

1. DiscountPolicy 라는 추상 클래스를 만드세요<sup>2</sup>.

<sup>2</sup>추상 클래스는 인스턴스화할 수 없는 클래스입니다. 추상 메서드는 추상 클래스에서만 선언할 수 있습니다.

2. 이 추상 클래스는 모든 할인 정책이 공통으로 가져야 할 `calculateDiscount(int price)` 라는 추상 메서드를 포함해야 합니다<sup>3</sup>.
3. 1단계에서 만든 `RateDiscountPolicy` 와 `AmountDiscountPolicy` 가 `DiscountPolicy` 를 상속(`extends`)받도록 수정하세요<sup>4</sup>.
4. 각 클래스의 생성자에서 할인율(%)이나 할인액(원)을 설정할 수 있도록 코드를 수정합니다<sup>5</sup>.

<sup>3</sup>추상 메서드는 추상 클래스에서만 선언할 수 있습니다.

<sup>4</sup>상속은 부모 클래스의 속성과 메서드를 자식 클래스에 물려주는 것을 의미합니다.

<sup>5</sup>생성자는 클래스의 인스턴스를 생성할 때 호출되는 메서드입니다.

### 1.3 오버로딩(Overloading) 추가 및 전체 코드 완성

이제 시스템이 고도화되어, “특별 회원”에게는 추가 할인을 제공하는 기능이 필요해졌습니다. 일반 회원은 기본 할인만, 특별 회원은 기본 할인 + 추가 할인을 적용합니다.

1. `RateDiscountPolicy` 와 `AmountDiscountPolicy` 클래스에 메서드 오버로딩을 적용하세요.
2. 기존 `calculateDiscount(int price)` 메서드는 그대로 둡니다. (일반 회원용)
3. `calculateDiscount(int price, boolean isSpecialMember)` 라는 새로운 메서드를 추가합니다. 이 메서드는 `isSpecialMember` 가 `true` 일 경우, 기본 할인에 “추가 할인”을 더 적용합니다.
4. `RateDiscountPolicy` : 5% 추가 할인
5. `AmountDiscountPolicy` : 1000원 추가 할인
6. `Main` 클래스에서 `RateDiscountPolicy` 와 `AmountDiscountPolicy` 객체를 생성하고, 상속과 오버로딩이 잘 동작하는지 테스트하는 코드를 완성하세요.