

## Soyutlama (Abstraction)

Nesne yönelimli programlamada Soyutlama (Abstraction) ilkesi, eğer bir sınıf için nesne üretmek mantıksız geliyorsa o sınıf soyutlanabilir. Alt sınıfların ortak özelliklerini ve işlevlerini taşıyan ancak henüz bir nesnesi olmayan bir üst sınıf oluşturmak istenirse bir soyut (abstract) üst sınıf oluşturulur.

Soyutlama, bir sınıfa veya metoda temel görevlerin tanımlanması, detayların ise tanımlanmaması demektir. Temel olarak bir soruna ait çözüme giderken kullanılacak yöntemlerin, ilk etapta daha genel basit ve soyut bir tanımını yapmaktır.

### "abstract" Anahtar Kelimesi ve Soyut Sınıf Kavramı (Abstract Class)

Soyutlama kavramı sınıfın içindeki iç işleyişi dışarıdan izole etmek, yani gizlemektir. Örneğin: bilgisayarı kullanırken çoğu kullanıcı bilgisayarın iç işleyişinden haberi olmaz. Hafızanın işlemciyle haberleşmesi, işlemler arası senkronizasyon, klavyeden girilen değerlerin ekrana yansması gibi birçok işlemin detayı kullanıcılardan gizlenmiş durumdadır. Kullanıcılar sadece bu fonksiyonları veya işlevleri bir arayüz vasıtasıyla çağırıp kullanmaktadır. İç detaylarına müdahale etmemektedir.

Aynı şekilde Java'da sınıflarımızı tasarlarken bazı fonksiyonların ve işlevlerin sadece sınıf içinde kalması, dış dünyada bu sınıftan nesneleri kullanan kişilerin bu iç fonksiyonları bilemelerine gerek yoktur. Örneğin: KDV tutarını hesaplayan fonksiyonun sınıf içinde kullandığı birçok başka fonksiyon olabilir. Bu fonksiyonların sınıf dışına açılmasının bir anlamı yoktur. Sadece miktarı verip o miktara göre KDV tutarını hesaplayacak bir dış fonksiyon yeterlidir. Yazılım dünyasında bu nedenle soyutlama kavramı yazılım tasarımında önemli bir kavramdır. Soyutlama yapabilmek için "abstract" anahtar kelimesi, "interface" gibi yapılar bizlere yardımcı olmaktadır.

Soyutlama için Java'da iki yöntem mevcuttur:

- "interface" tanımlamak
- "abstract" sınıf tanımlamak

### Soyut Sınıf (Abstract Class)

"abstract" anahtar kelimesi ile tanımlanan sınıflardır. Sınıfın içinde soyut ("abstract") metotlar veya normal fonksiyonlar tanımlanabilir. Soyut sınıflardan "new" anahtar kelimesi ile bir nesne oluşturulamaz.

#### Soyut Sınıf Özellikleri:

- "abstract" anahtar kelimesi ile tanımlanmış olması gerekiyor.

- Soyut veya soyut olmayan fonksiyonlar tanımlanabilir.
- Soyut sınıflardan "new" anahtar kelimesi ile nesne oluşturulamaz.
- Kurucu metodu ve static fonksiyonlar tanımlanabilir.
- "final" kelimesi ile tanımlanmış fonksiyonları içerebilir. Bu final fonksiyonlar alt sınıflarda ezilemezler (override).

```
// abstract sınıf örneği
public abstract class Doping {

    protected double price;
    protected double[] taxes;

    public double[] getTaxes() {
        return taxes;
    }

    public void setTaxes(double[] taxes) {
        this.taxes = taxes;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    // soyut metot örneği
    public abstract double calculate();
}
```

Yukarıda soyut bir sınıf tanımladık. "abstract" kelimesi ile sınıf tanımladık, ayrıca sınıfın içinde "calculate" isimli "abstract" metot tanımladık. Aynı zamanda soyut olmayan metotlar da tanımladık. Senaryomuzda bir e-ticaret sisteminde "Doping" tipinde ek ürünler olduğunu düşünelim. İlan tarihini güncelleyen bir doping çeşidimiz olsun, bir de üst sırada çıkmanızı sağlayan bir doping olsun. Bu iki alt sınıfta "Doping" isimli sınıftan kalıtım alarak belli özellikleri kendilerine alsınlar. Fakat, her dopingin ücret hesaplama yöntemi birbirinden farklı olabilir. Ayrıca, her dopingin mutlaka fiyat hesaplama fonksiyonu olmalıdır.

Yukarıdaki durumda "abstract" sınıf tanımlayıp diğer doping çeşitleri bu ATA sınıftan kalıtım alacaklardır. "calculate" isimli "abstract" metodu, "metod ezme" (overriding) yöntemiyle ezip metodun içeriğini kendilerine göre dolduracaklardır. Alt sınıflardaki diğer özellikler soyutlama tekniğiyle dış dünyadan gizlenecektir. Dış dünyadan dopingi kullanmak isteyen başka bir sınıf veya kod parçası doping nesnesi üzerindeki "calculate" fonksiyonunu çağırıp fiyatı hesaplayacaktır. Diğer iç hesaplama ve çalışma detaylarını bilmeyecektir.

```
public class TopOfListDoping extends Doping {

    public TopOfListDoping(double price)
    {
```

```

        super.setPrice(price);
    }

    // "Doping" soyut sınıfından kalıtımla gelen, "calculate" isimli soyut metodu
    // metot ezmesi yöntemiyle alt sınıf kendi ihtiyacına göre dolduruyor.
    // "TopOfList" isimli doping tipinde vergiler olmadığı için komisyon oranı
    // eklenip ücret hesaplanıyor. Fakat, başka doping çeşitlerinde hesaplama farklı
    // olabilir.
    @Override
    public double calculate() {

        return super.getPrice() + super.getPrice() * 0.35;
    }
}

public class UptodateDoping extends Doping {

    public UptodateDoping(double price, double[] taxes) {
        super.setPrice(price);
        super.setTaxes(taxes);
    }

    // "Doping" soyut sınıfından kalıtımla gelen, "calculate" isimli soyut metodu
    // metot ezmesi yöntemiyle alt sınıf kendi ihtiyacına göre dolduruyor.
    // "UptodateDoping" isimli doping tipinde vergiler fiyata dahil olduğu için
    // komisyon oranı eklenip ve vergiler hesaplanıp ücret belirleniyor.
    // Görüldüğü gibi her doping çeşidinin fiyat hesaplama yöntemleri birbirinden
    // farklıdır. Soyutlama ile sınıflara ait iç çalışma detayları gizlenmiş oluyor.
    // Doping tiplerinde sadece "calculate" isimli fonksiyonu dış dünyaya açtık.
    // Diğer tüm fonksiyonlar ve özellikler sınıf içinde kaldı.
    @Override
    public double calculate() {

        return calculateTaxes() + commisionRate();
    }

    private double calculateTaxes() {

        double totalTaxValue = 0;
        for(int i=0; i < super.getTaxes().length; i++) {
            totalTaxValue += super.getTaxes()[i];
        }
        return totalTaxValue;
    }

    private double commisionRate() {
        return super.getPrice() * 0.2;
    }
}

```

"Doping" soyut sınıfından kalıtımla gelen, "calculate" isimli soyut metodu metot ezmesi yöntemiyle alt sınıf kendi ihtiyacına göre dolduruyor. "UptodateDoping" isimli doping tipinde vergiler fiyata dahil olduğu için komisyon oranı eklenip ve vergiler hesaplanıp ücret belirleniyor. Görüldüğü gibi her doping çeşidinin fiyat hesaplama yöntemleri birbirinden farklıdır. Soyutlama ile sınıflara ait iç çalışma detayları gizlenmiş oluyor. Doping tiplerinde

sadece "calculate" isimli fonksiyonu dış dünyaya açtık. Diğer tüm fonksiyonlar ve özellikler sınıf içinde kaldı.