



Bölüm 4: Soyut Sınıflar

JAVA ile Nesne Yönelimli Programlama



Soyut Sınıflar (Abstract Classes)

- abstract olarak tanımlanır.
- Soyut sınıflar, soyut veya soyut olmayan metodlar içerebilir.

```
public abstract class Sekil {  
    // Soyut olmayan metot  
    void tasi(int x, int y) {  
        // Metot içeriği  
    }  
    // Soyut yöntem (alt sınıflarca implemente edilmeli)  
    abstract void ciz();  
}
```



Soyut Sınıflar

- Soyut sınıfların nesneleri oluşturulamaz.
- Başka bir sınıf tarafından kalıtım yoluyla türetilebilir.
- Alt sınıf soyut metodları gerçeklemek zorundadır.
- Ortak durumları ve davranışları birleştirir, tekrar kullanım sağlar.
- Alt sınıflar, soyut metotları kendi ihtiyaçlarına göre özelleştirebilir.
- Yapıcı fonksiyonlara (constructor) sahip olabilirler.



Soyut Metotlar

- Gerçekleştirim (implementasyon) olmadan tanımlanır.
- Süslü parantez içermeyen ve noktalı virgülle biten bir tanımlamadır.
- `public abstract void ciz();`
- Soyut sınıflar genellikle kalıtım hiyerarşisinde yer alır.



Soyut Metotlar

```
public abstract class Sekil {  
    // Soyut metot  
    public abstract void ciz();  
}
```

```
public class Cember extends Sekil {  
    // ciz metodunu implemente etmek zorunlu  
    public void ciz() {  
        // Çemberi çiz  
    }  
}
```



Arayüzler (Interfaces)

- Sistemin belirli işlevlere sahip olacağını garanti eden sözleşmedir.
- İki sistem arasındaki entegrasyon noktasını temsil eder.
- Arayüz tanımlamak için interface anahtar kelimesi kullanılır.
- Metotlar public ve abstract olmalıdır.
- public static final değişkenleri (sabitleri) içerebilir.



Arayüzler

```
public interface GeometrikSekil {  
  
    // Soyut yöntem (public abstract)  
    void çiz();  
  
    // Sabit (public static final)  
    int KENAR_SAYISI = 4;  
}
```



Arayüz Gerçekleme

- Bir sınıf, bir arayüzü implements anahtar kelimesiyle gerçekler.
- Arayüzdeki tüm metotları gerçeklemeli veya soyut olarak tanımlamalıdır.
- Bir sınıf birden fazla arayüzü gerçekleyebilir.

```
public class Kare implements GeometrikSekil {  
    // ciz metodunu implemente etmek zorunlu  
    public void ciz() {  
        // Kareyi çiz  
    }  
}
```




Arayüzden Kalıtım

- Bir arayüz, başka bir arayüzü extends anahtar kelimesi ile genişletebilir.

```
public interface KareSekli extends GeometrikSekil {  
    // Ek metodlar veya sabitler eklenebilir  
    void ozelFonksiyon();  
}
```



Arayüz Soyut Sınıf Karşılaştırma

- Arayüz
 - Yalnızca soyut metotlar içerir.
 - Birden çok arayüzden türeyebilir.
 - Çoklu kalıtım özellikleri sağlar.
- Soyut Sınıf
 - Hem soyut hem de somut metotları içerebilir.
 - Tek bir soyut sınıftan türetilir.
 - Ortak niteliklere sahip sınıfları gruplamak için kullanılır.



SON