Laboratuvar Çalışması

`this` Anahtar Sözcüğü ve Kalıtım (Inheritance)

Arş. Gör. Ömer Miraç KÖKÇAM
TF Yazılım Mühendisliği – YMT1102

5 Mayıs 2026

Yönerge

- Bu lab çalışması **2 sorudan** oluşmaktadır. Her iki soruyu da çözmeniz beklenmektedir.
- Kodlarınızı bir IDE'de (IntelliJ IDEA, Eclipse veya VS Code) yazıp **derleyip çalıştırın**.
- Soru 1 için `Hasta.java` ve `TestHasta.java`; Soru 2 için `Kisi.java`, `Personel.java`, `Doktor.java` ve `TestHastane.java` dosyalarını oluşturun.

Soru 1: `this` Anahtar Sözcüğü – Hasta Kayıt Sistemi

Bir hastane acil servisinde hasta kayıt sistemi tasarlıyorsunuz. Acil durumlarda hasta her zaman tam bilgiyle kaydedilemez; sisteme hasta bazen sadece adıyla, bazen adı ve TC kimliğiyle, bazen de tüm bilgileriyle eklenebilmelidir.

Yapılacaklar:

1. `Hasta` sınıfını aşağıdaki alanlarla oluşturun: `String ad`, `String tcKimlik`, `String telefon`, `String kanGrubu`.
2. `this(...)` kullanarak **üç ayrı constructor** yazın (constructor chaining):
   - Yalnızca `ad` parametresi alan constructor: `tcKimlik`, `telefon` ve `kanGrubu` alanlarını `"Bilinmiyor"` olarak iki parametreli constructor'a yönlendirsin.
   - `ad` ve `tcKimlik` parametresi alan constructor: `telefon` ve `kanGrubu` alanlarını `"Bilinmiyor"` olarak tam constructor'a yönlendirsin.
   - Dört parametreyi de alan tam constructor: `this.alan = parametre` biçiminde tüm alanları doldursun.
3. `void bilgileriGoster()` metodu yazın; hastanın tüm alanlarını ekrana yazsın.
4. `TestHasta` sınıfında `main` metodu içinde **üç farklı hasta nesnesi** oluşturun (her constructor için bir tane) ve `bilgileriGoster()` ile ekrana yazdırın.

Beklenen Çıktı (örnek)

```
Ad: Ahmet Yilmaz | TC: Bilinmiyor | Tel: Bilinmiyor | Kan: Bilinmiyor
Ad: Ferhat Ucar  | TC: 12345678901 | Tel: Bilinmiyor | Kan: Bilinmiyor
Ad: Ayse Demir   | TC: 98765432100 | Tel: 05321234567 | Kan: A Rh+
```

Hatırlatma

`this(...)` çağrısı constructor'ın *ilk satırı* olmak zorundadır; aksi hâlde derleme hatası alırsınız. Bir constructor içinde hem `this(...)` hem `super(...)` birlikte kullanılamaz.

Düşündürücü Sorular

1. `this.ad = ad;` satırında sol taraftaki `this.ad` ile sağ taraftaki `ad` farklı şeyleri mi ifade ediyor? Açıklayın.
2. Tek parametreli constructor içinde `this(ad, "Bilinmiyor", "Bilinmiyor", "Bilinmiyor")` yazdığınızda aslında hangi constructor çalışıyor? Neden bu yaklaşım *kod tekrarını* önlüyor?

Soru 2: Kalıtım (Inheritance) – Hastane Personel Hiyerarşisi

Hastane sisteminde tüm çalışanlar bir `Kisi` üst sınıfından türemektedir. `Personel` sınıfı `Kisi`'den, `Doktor` sınıfı ise `Personel`'den türeyecektir. Bu üç katmanlı hiyerarşiyi aşağıdaki adımları izleyerek oluşturun.

Yapılacaklar:

1. `Kisi` sınıfını oluşturun. Alanlar: `String ad`, `String tcKimlik`.
   - İki parametreyi alan bir constructor yazın; `this.alan = parametre` biçiminde doldurun.
   - `void bilgileriGoster()` metodu yazın; `ad` ve `tcKimlik` alanlarını yazdırsın.
2. `Personel` sınıfını `Kisi`'den türetin (`extends Kisi`). Ek alanlar: `String sicilNo`, `String bolum`.
   - Constructor'ın ilk satırında `super(ad, tcKimlik)` ile üst sınıfı başlatın.
   - Kendi alanlarını `this` ile doldurun.
   - `bilgileriGoster()` metodunu `@Override` ile ezip önce `super.bilgileriGoster()` çağırın, ardından `sicilNo` ve `bolum` alanlarını ekleyin.
3. `Doktor` sınıfını `Personel`'den türetin (`extends Personel`). Ek alanlar: `String uzmanlik`, `String diplomaNo`.
   - Constructor'ın ilk satırında `super(...)` ile `Personel`'in constructor'ını çağırın.
   - Kendi alanlarını doldurun.
   - `bilgileriGoster()` metodunu override edin; `super.bilgileriGoster()` ardından `uzmanlik` ve `diplomaNo` alanlarını ekleyin.
4. `TestHastane` sınıfında `main` metodu içinde:
   - Bir `Kisi`, bir `Personel` ve bir `Doktor` nesnesi oluşturun.
   - Her birinde `bilgileriGoster()` çağırın.
   - `instanceof` operatörü ile şu üç ifadenin sonucunu ekrana yazdırın ve neden böyle olduğunu kodun yanına yorum satırı olarak açıklayın: `doktor instanceof Personel`, `doktor instanceof Kisi`, `personel instanceof Doktor`.

Beklenen Çıktı (örnek)

```
--- Kisi ---
Ad: Ferhat Ucar | TC: 11111111111

--- Personel ---
Ad: Ahmet Yilmaz | TC: 22222222222
Sicil: P001 | Bolum: Yonetim

--- Doktor ---
Ad: Ali Can | TC: 33333333333
Sicil: D001 | Bolum: Kardiyoloji
Uzmanlik: Kalp Hastaliklari | Diploma: DR-2018-042

instanceof sonuclari:
doktor instanceof Personel : true
doktor instanceof Kisi     : true
personel instanceof Doktor  : false
```

Hatırlatma

`super(...)` çağrısı da tıpkı `this(...)` gibi constructor'ın *ilk satırı* olmak zorundadır. `@Override` yazmak zorunlu değildir; ancak yazarsanız Java, metodun gerçekten bir üst sınıf metodunu ezdiğini derleme anında doğrular.

Düşündürücü Sorular

1. `Doktor` sınıfının constructor'ında `Kisi`'nin alanlarına (`ad`, `tcKimlik`) doğrudan `this.ad = ...` ile değer atamayı deneyin. Ne oluyor? Neden `super(...)` kullanmak *zorunlu*?
2. `Doktor` nesnesi üzerinde `bilgileriGoster()` çağrıldığında üç sınıfın çıktısı sırayla ekrana geldi. Bu sırayı sağlayan mekanizma nedir?
3. Java'da `class Doktor extends Personel, Kisi` yazmayı deneyin. Hata alıyor musunuz? Java neden tek kalıtıma izin veriyor?