

# 3. Karşılaştırma Operatörleri

## Örnek 1: Ehliyet Yeterlilik Kontrolü

**Amaç:** `>=` (büyük veya eşit) ve `<` (küçük) operatörlerini gerçek hayattan bir senaryoda kullanmak.

- **Senaryo:** Türkiye'de ehliyet alabilmek için yaşı 18 veya daha büyük olması gerekiyor. Bir kişinin yaşını kontrol edip ehliyet alıp alamayacağını bulan bir program yazalım.
- **İstenenler:**
  1. `EHLIYET_YASI` adında bir sabit oluşturun ve değerini `18` yapın.
  2. `basvuraninYasi` adında bir değişken oluşturun ve bir yaş değeri atayın (örneğin `20`).
  3. `ehliyetAlabilirMi` adında bir değişken oluşturun. Bu değişkene, başvuranın yaşının ehliyet yaşından büyük veya eşit olup olmadığını karşılaştırarak çıkan sonucu (`true` veya `false`) atayın.
  4. Sonucu ekrana "Ehliyet alabilir mi?: " şeklinde yazdırın.
  5. `basvuraninYasi` değerini `16` yapıp programı tekrar çalıştırın ve sonucu gözlemleyin.

- **Beklenen Çıktı (Yaş 20 iken):**

```
Ehliyet alabilir mi?: true
```

- **Beklenen Çıktı (Yaş 16 iken):**

```
Ehliyet alabilir mi?: false
```

## Örnek 2: Oyun Rekoru Kırıldı mı?

**Amaç:** `>` (büyük), `==` (eşit) ve `!=` (eşit değil) operatörlerini eğlenceli bir senaryoda kullanmak.

- **Senaryo:** Bir oyundaki en yüksek skor 15.000 puandır. Oyuncu oyunu bitirdiğinde yeni bir skor elde ediyor. Bu yeni skorun rekoru kırıp kırmadığını, rekora eşit olup olmadığını ve rekordan farklı olup olmadığını kontrol eden bir program yazalım.
- **İstenenler:**
  1. `enYuksekSkor` adında bir değişken oluşturun ve değerini `15000` yapın.
  2. `oyuncununSkoru` adında bir değişkeni oluşturun ve bir skor atayın (örneğin `17500`).
  3. `rekorKirildiMi` (`>`), `rekorEgalendiMi` (`==`) ve `skorlarFarkliMi` (`!=`) adında üç ayrı `boolean` değişkeni oluşturun.
  4. Her bir boolean değişkenine, uygun karşılaştırma işleminin sonucunu atayın.
  5. Tüm sonuçları ekrana ayrı ayrı yazdırın.
  6. `oyuncununSkoru` değerini `14000` yapıp sonuçları tekrar gözlemleyin.
- **Beklenen Çıktı (Skor 17500 iken):**

```
Rekor Kırıldı mı?: true
Rekor Egalendi mi?: false
Skorlar Farklı mı?: true
```

### Örnek 3: Alışveriş Bütçesi

**Amaç:** `<=` (küçük veya eşit) operatörünü kullanarak pratik bir bütçe kontrolü yapmak.

- **Senaryo:** Markete gittin ve almak istediğin bir ürün var. Cüzdanındaki paranın bu ürünü almaya yetip yetmediğini kontrol eden bir program yaz.

- **İstenenler:**

1. `cuzdandakiPara` adında bir `double` değişkeni oluştur ve bir miktar ata (örneğin `85.50` ).
2. `urunFiyati` adında bir `double` değişkeni oluştur ve bir fiyat ata (örneğin `79.99` ).
3. `paramYeterliMi` adında bir `boolean` değişkeni oluştur.
4. Ürün fiyatının cüzdanındaki paradan küçük veya eşit olup olmadığını karşılaştırarak sonucu `paramYeterliMi` değişkenine ata.
5. Sonucu "Bu ürünü alabilir miyim?: " şeklinde ekrana yazdır.
6. `urunFiyati` 'nı `90.0` yaparak sonucu tekrar gözlemle.

- **Beklenen Çıktı (Fiyat 79.99 iken):**

```
Bu ürünü alabilir miyim?: true
```

## Bonus: ATM PIN Kodu Kontrolü

**Amaç:** Birden fazla karşılaştırma operatörünü tek bir senaryoda birleştirmek.

- **Senaryo:** Çok basit bir ATM ekranı simülasyonu yapıyoruz. Kullanıcının girdiği PIN kodunun doğru olup olmadığını kontrol edeceğiz. Doğru PIN 4 haneli "1923"tür.
- **İstenenler:**
  1. `dogruPIN` adında bir `final int` sabiti oluştur ve değerini `1923` yap.
  2. `girilenPIN` adında bir `int` değişkeni oluştur ve bir değer ata (örneğin `1234`).
  3. `pinDogruMu` adında bir `boolean` oluştur ve `girilenPIN` 'in `dogruPIN` 'e eşit olup olmadığını ( `==` ) kontrol et.
  4. `pinYanlisMi` adında bir `boolean` oluştur ve `girilenPIN` 'in `dogruPIN` 'den farklı olup olmadığını ( `!=` ) kontrol et.
  5. Tüm sonuçları ekrana yazdır.
  6. `girilenPIN` değerini `1923` yaparak programı tekrar çalıştır ve sonuçları incele.
- **Beklenen Çıktı (PIN 1234 iken):**

```
Girilen PIN Doğru mu?: false
Girilen PIN Yanlış mı?: true
```