Bölüm 7: Koşullu İfadeler

Bilgisayar ve Programlamaya Giriş

Akış | if-else | else if | switch | Örnekler

Sercan KÜLCÜ · Giresun Üniversitesi

© Sercan Külcü. Tüm hakları saklıdır. · Sunumun etkileşimli HTML sürümü: https://sercankulcu.github.io/files/computer/slides/Bolum_07_Kosullu_Ifadeler.html

GİRİŞ

Öğrenme Çıktıları

Akış

Sıralı, seçmeli ve tekrarlı akışı ayırt etmek.

Koşul

Karşılaştırma ve mantık işleçleriyle koşul yazmak.

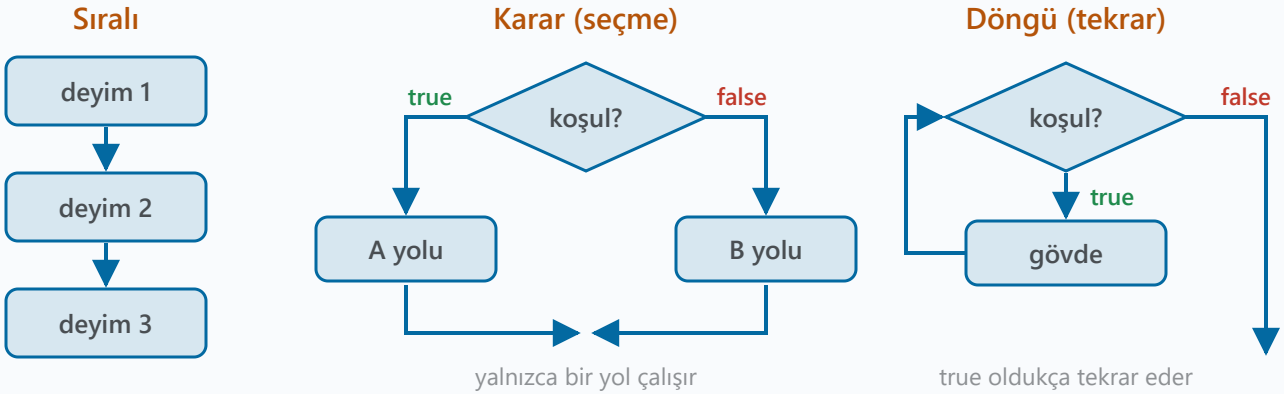
Karar

if, else if ve else yapılarını doğru sırada kurmak.

switch

Ayrık seçeneklerde switch kullanmak.

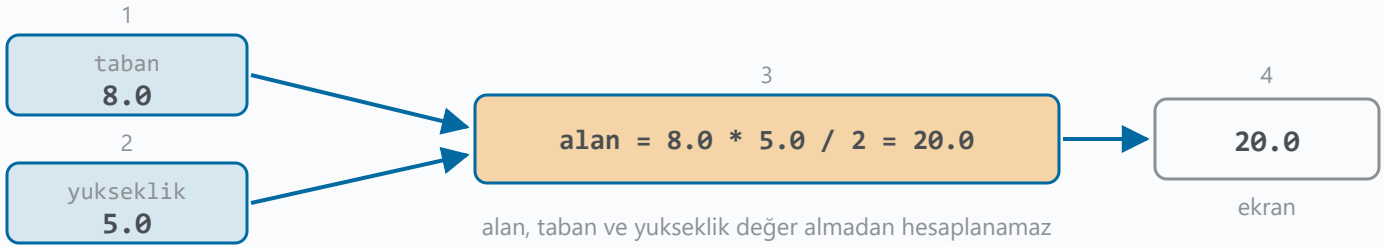
Üç Temel Akış



- Sıralı yapı deyimleri yazıldığı sırada yürütür; karar yapısı koşula göre farklı yol seçer.
- Döngü yapısı bir kod bloğunu tekrarlar.
- Bu üç yapı her algoritmanın temel yürütme modelini oluşturur.

Sıralı İşlemler

```
double taban = 8;
double yukseklik = 5;
double alan = taban * yukseklik / 2;
System.out.println(alan);
```



- Her deyim bir öncekinin ürettiği değeri kullanabilir.
- Sıra değiştirilirse tanımsız veya yanlış değer oluşabilir.

Sıralı İşlemler Adım Adım



taban = 8 → double kutuya 8.0 olarak yazılır (genişletme).

Adım 1 / 5

```
▶ double taban = 8;
2 double yukseklik = 5;
3 double alan = taban * yukseklik / 2;
4 System.out.println(alan);
```

DEĞİŞKENLER

`taban = 8.0`

ÇIKTI

Sıralı İşlemler Adım Adım



yukseklik = 5.0.

Adım 2 / 5

```
1 double taban = 8;
▶ double yukseklik = 5;
3 double alan = taban * yukseklik / 2;
4 System.out.println(alan);
```

DEĞİŞKENLER

`taban = 8.0` `yukseklik = 5.0`

ÇIKTI

Sıralı İşlemler Adım Adım

Bellek (değişkenler)

taban	yukseklik
8.0	5.0
double	double

İfade

```
taban * yukseklik / 2
40.0 / 2
```

Soldan sağa: önce taban * yukseklik = 40.0.

Adım 3 / 5

```
1 double taban = 8;
2 double yukseklik = 5;
▶ double alan = taban * yukseklik / 2;
4 System.out.println(alan);
```

DEĞİŞKENLER

taban = 8.0 | yukseklik = 5.0

ÇIKTI

Sıralı İşlemler Adım Adım

Bellek (değişkenler)



40.0 / 2 = 20.0 → alan kutusuna yazılır. Bu satır önceki iki kutunun değerini **kullanır**.

Adım 4 / 5

```
1 double taban = 8;
2 double yukseklik = 5;
▶ double alan = taban * yukseklik / 2;
4 System.out.println(alan);
```

DEĞİŞKENLER

`taban = 8.0` `yukseklik = 5.0` `alan = 20.0`

ÇIKTI

Sıralı İşlemler Adım Adım

Bellek (değişkenler)

taban	yukseklik	alan
8.0	5.0	20.0
double	double	double

20.0 yazdırılır. Satır 3, satır 1'den önce olsaydı taban henüz tanımsız olurdu → derleme hatası.

Adım 5 / 5

```
1 double taban = 8;
2 double yukseklik = 5;
3 double alan = taban * yukseklik / 2;
▶ System.out.println(alan);
```

DEĞİŞKENLER

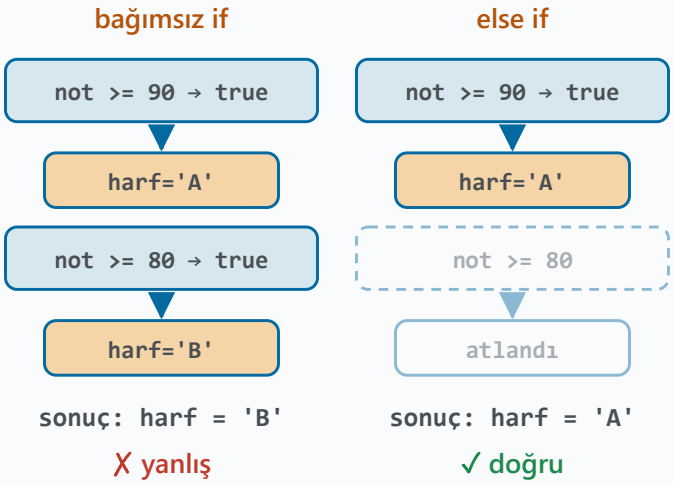
`taban = 8.0` `yukseklik = 5.0` `alan = 20.0`

ÇIKTI

```
20.0
```

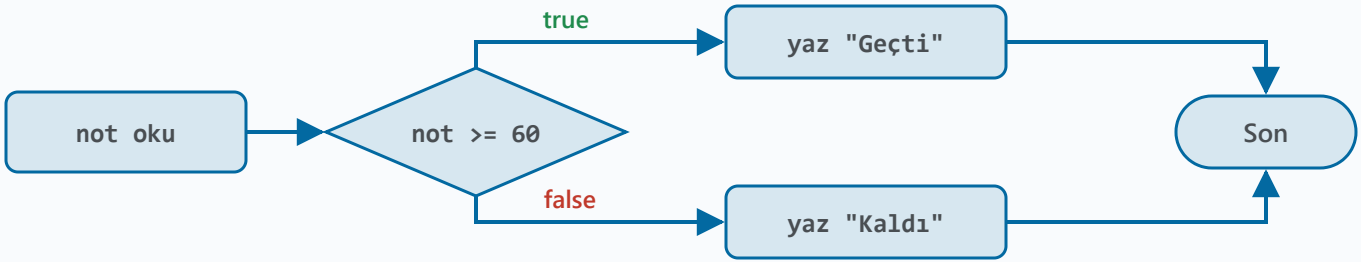
Karar Verme

- Koşul boolean değer üretir.
- if doğru yolu, else yanlış yolu tanımlar.
- Bağımsız if'lerin **hepsi** denetlenir; else-if zincirinde ilk doğru dal çalışınca gerisi **atlanır**.
- Örnek: not = 95. Sağda iki yazımın farklı sonucu.



KARARLAR

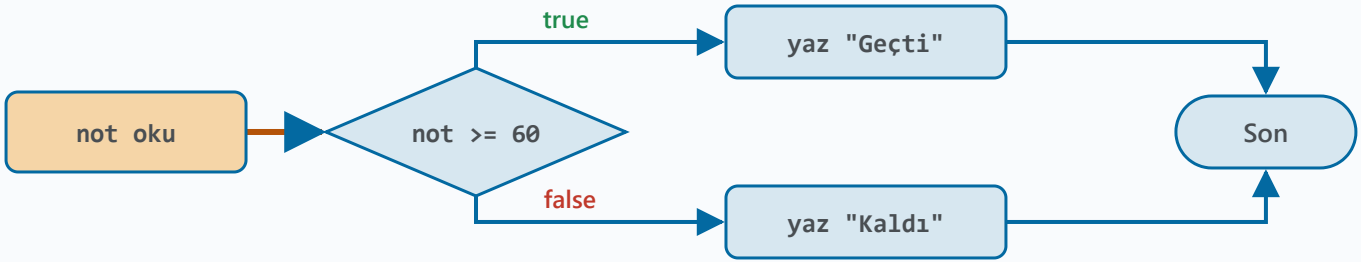
if-else Yapısı




```
if (not >= 60) {
    System.out.println("Geçti");
} else {
    System.out.println("Kaldı");
}
```

- Sınır değeri koşulun parçasıdır: 60 → Geçti.
- Her olası değer yalnızca **bir** dala gider.

if-else Adım Adım: not = 75, 60, 45



1. çalıştırma: kullanıcı **75** girer.

Adım 1 / 9

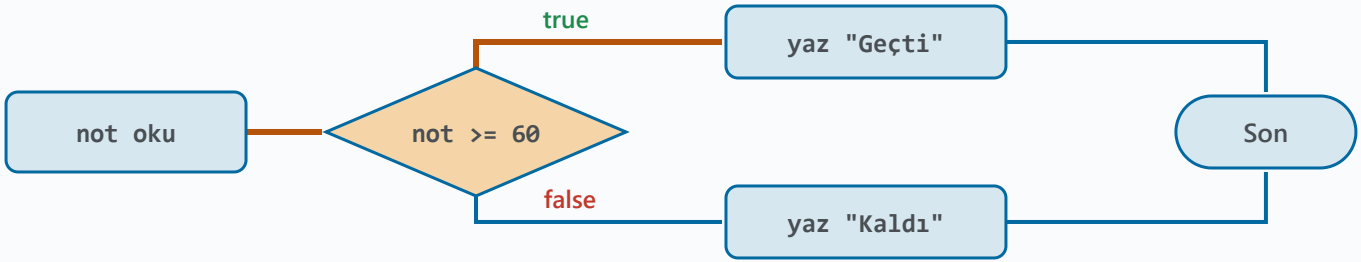
```
1 Scanner klavye = new Scanner(System.in);
▶ int not = klavye.nextInt();
3 if (not >= 60) {
4     System.out.println("Geçti");
5 } else {
6     System.out.println("Kaldı");
7 }
```

DEĞİŞKENLER

`not = 75`

ÇIKTI

if-else Adım Adım: not = 75, 60, 45



Koşul: not >= 60 → 75 >= 60 → **true**.

Adım 2 / 9

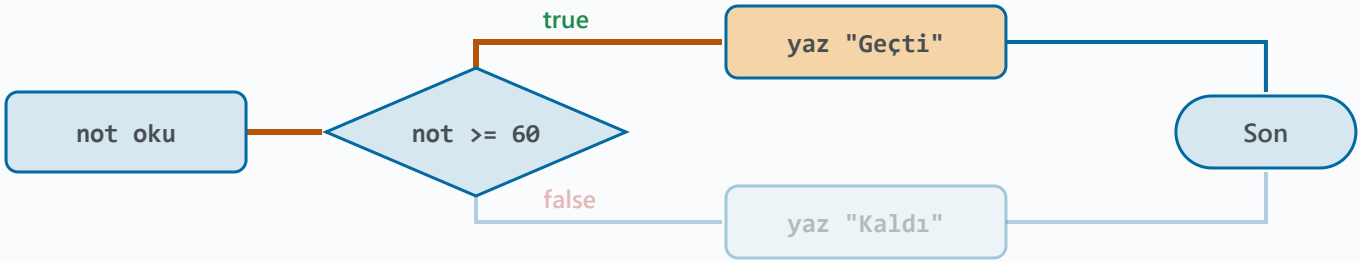
```
1 Scanner klavye = new Scanner(System.in);
2 int not = klavye.nextInt();
▶ if (not >= 60) {
4     System.out.println("Geçti");
5 } else {
6     System.out.println("Kaldı");
7 }
```

DEĞİŞKENLER

`not = 75`

ÇIKTI

if-else Adım Adım: not = 75, 60, 45



true dalı çalışır: **Geçti**. Diğer dal (else bloğu) atlanır.

Adım 3 / 9

```
1 Scanner klavye = new Scanner(System.in);
2 int not = klavye.nextInt();
3 if (not >= 60) {
▶     System.out.println("Geçti");
5 } else {
6     System.out.println("Kaldı");
7 }
```

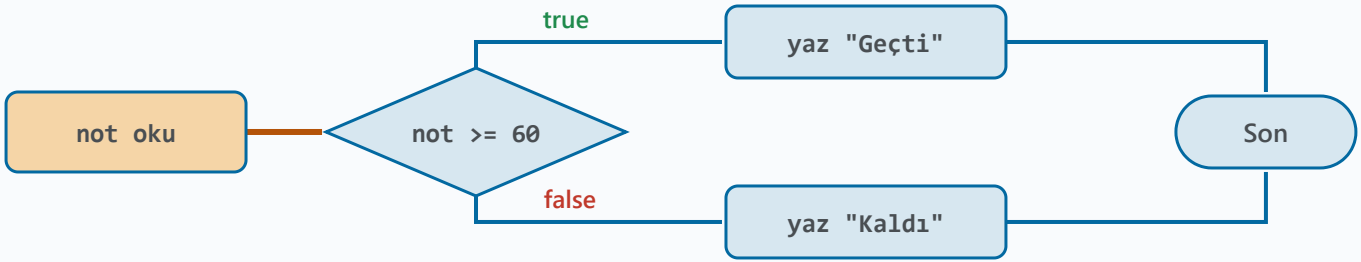
DEĞİŞKENLER

`not = 75`

ÇIKTI

```
Geçti
```

if-else Adım Adım: not = 75, 60, 45



2. çalıştırma: kullanıcı **60** girer.

Adım 4 / 9

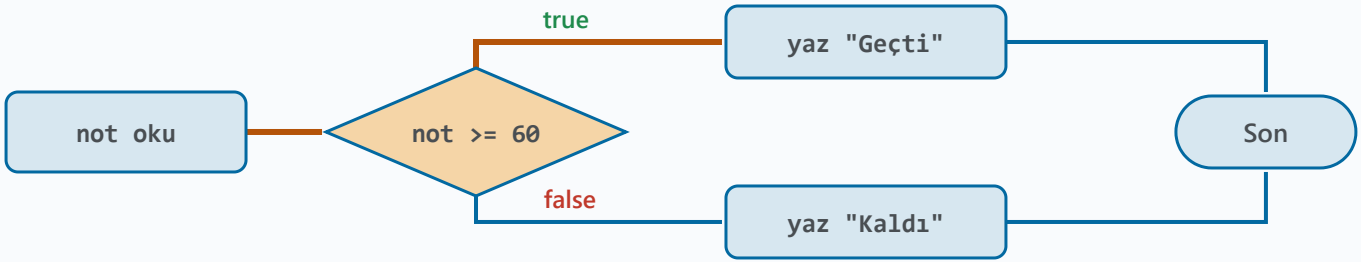
```
1 Scanner klavye = new Scanner(System.in);
▶ int not = klavye.nextInt();
3 if (not >= 60) {
4     System.out.println("Geçti");
5 } else {
6     System.out.println("Kaldı");
7 }
```

DEĞİŞKENLER

`not = 60`

ÇIKTI

if-else Adım Adım: not = 75, 60, 45



Koşul: not >= 60 → 60 >= 60 → **true**. Sınır değeri dahil: `>=` .

Adım 5 / 9

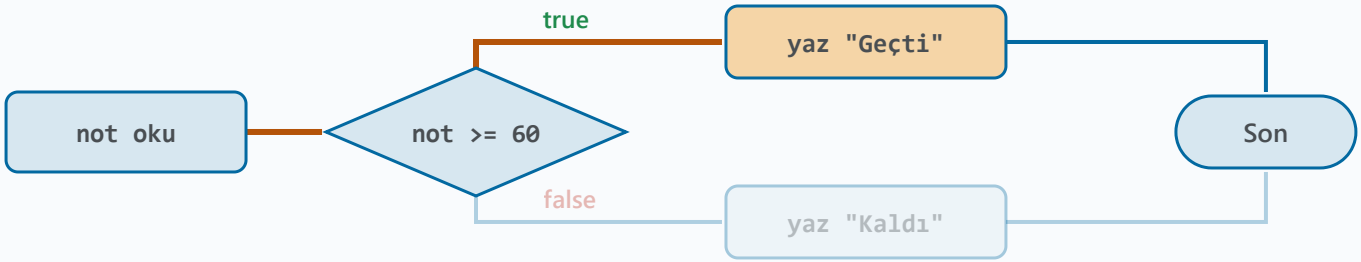
```
1 Scanner klavye = new Scanner(System.in);
2 int not = klavye.nextInt();
▶ if (not >= 60) {
4     System.out.println("Geçti");
5 } else {
6     System.out.println("Kaldı");
7 }
```

DEĞİŞKENLER

`not = 60`

ÇIKTI

if-else Adım Adım: not = 75, 60, 45



true dalı çalışır: **Geçti**. Diğer dal (else bloğu) atlanır.

Adım 6 / 9

```
1 Scanner klavye = new Scanner(System.in);
2 int not = klavye.nextInt();
3 if (not >= 60) {
▶     System.out.println("Geçti");
5 } else {
6     System.out.println("Kaldı");
7 }
```

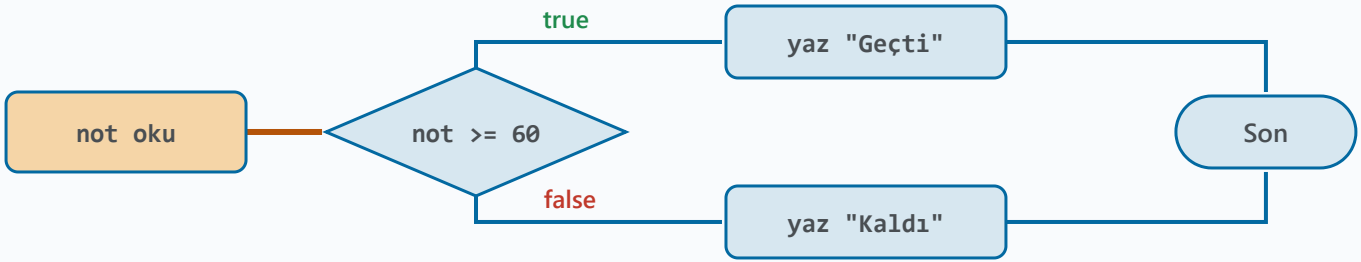
DEĞİŞKENLER

`not = 60`

ÇIKTI

```
Geçti
```

if-else Adım Adım: not = 75, 60, 45



3. çalıştırma: kullanıcı **45** girer.

Adım 7 / 9

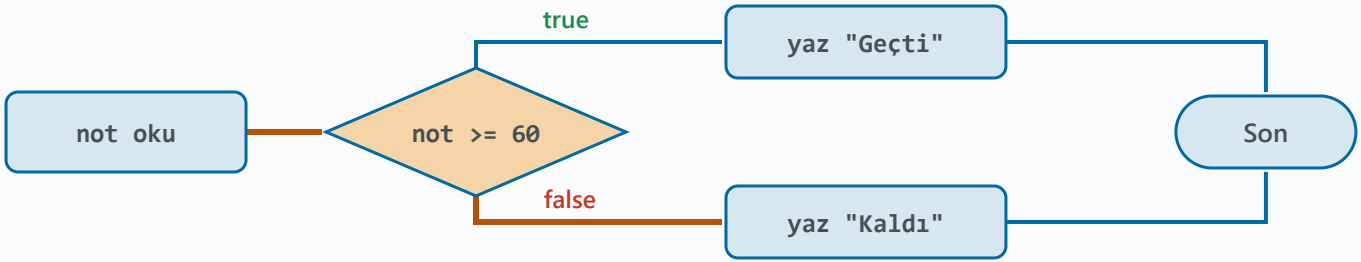
```
1 Scanner klavye = new Scanner(System.in);
▶ int not = klavye.nextInt();
3 if (not >= 60) {
4     System.out.println("Geçti");
5 } else {
6     System.out.println("Kaldı");
7 }
```

DEĞİŞKENLER

`not = 45`

ÇIKTI

if-else Adım Adım: not = 75, 60, 45



Koşul: not >= 60 → 45 >= 60 → **false**.

Adım 8 / 9

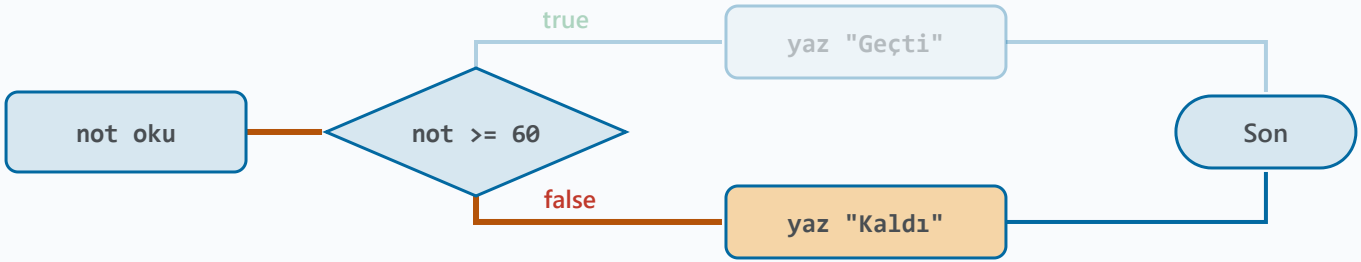
```
1 Scanner klavye = new Scanner(System.in);
2 int not = klavye.nextInt();
▶ if (not >= 60) {
4     System.out.println("Geçti");
5 } else {
6     System.out.println("Kaldı");
7 }
```

DEĞİŞKENLER

not = 45

ÇIKTI

if-else Adım Adım: not = 75, 60, 45



false dalı çalışır: **Kaldı**. Diğer dal (if bloğu) atlanır.

Adım 9 / 9

```
1 Scanner klavye = new Scanner(System.in);
2 int not = klavye.nextInt();
3 if (not >= 60) {
4     System.out.println("Geçti");
5 } else {
▶     System.out.println("Kaldı");
7 }
```

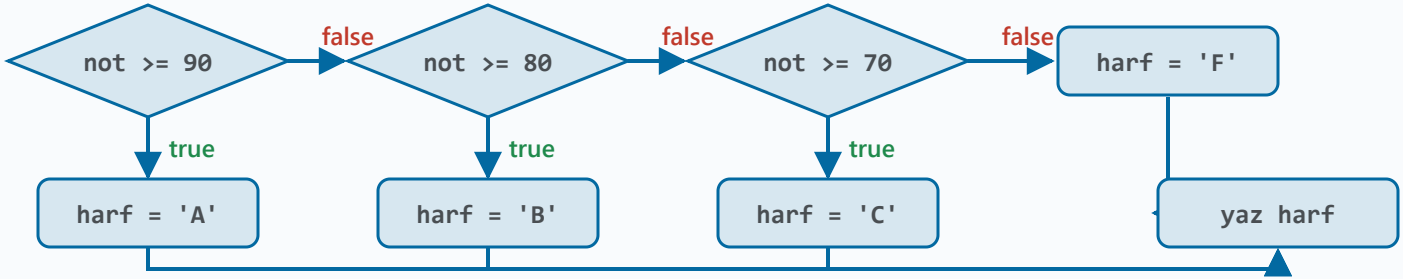
DEĞİŞKENLER

`not = 45`

ÇIKTI

```
Kaldı
```

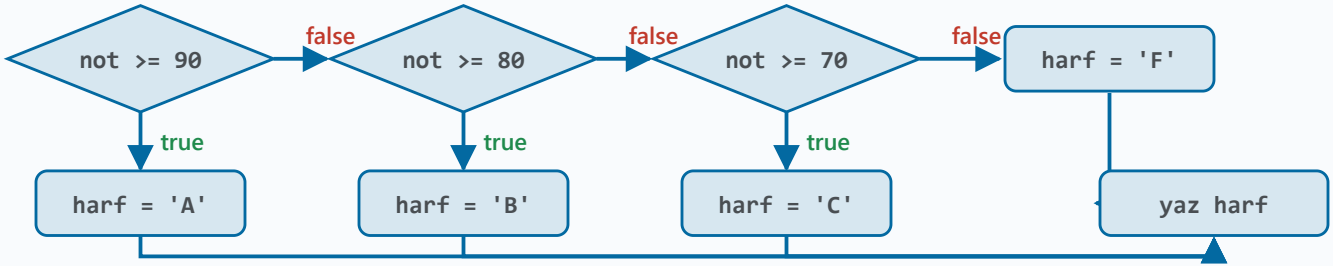
Çok Yollu Karar: else-if




```
if (not >= 90) {
    harf = 'A';
} else if (not >= 80) {
    harf = 'B';
} else if (not >= 70) {
    harf = 'C';
} else {
    harf = 'F';
}
```

- Koşullar yukarıdan aşağı değerlendirilir; **ilk true** dal çalışır, gerisi atlanır.
- Yüksek eşikler önce yazılır: `not >= 70` başa yazılsaydı 95 de C alırdı.
- else hiçbir koşul tutmadığında çalışır.

else-if Zinciri: not = 75, 95, 50



1. çalıştırma: not = **75**. harf bildirildi, henüz değeri yok.

Adım 1 / 16

```
▶ int not = klavye.nextInt();
▶ char harf;
3 if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

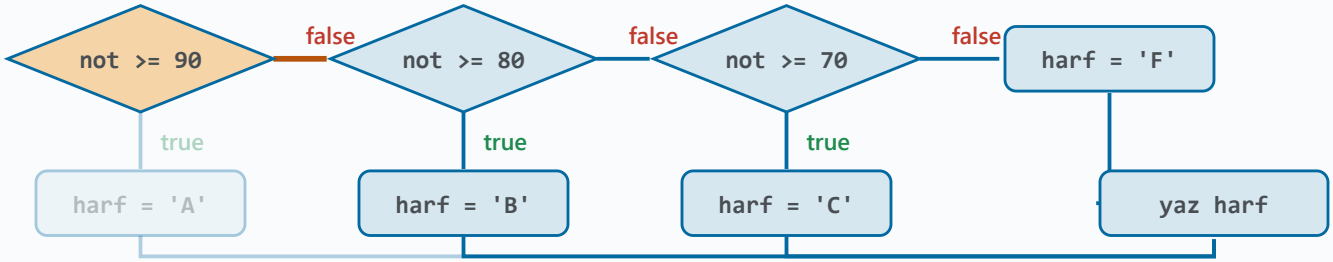
DEĞİŞKENLER

`not = 75` `harf = ?`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



1. koşul: not >= 90 → 75 >= 90 → **false**. Sonraki koşula geçilir.

Adım 2 / 16

```
1 int not = klavye.nextInt();
2 char harf;
▶ if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

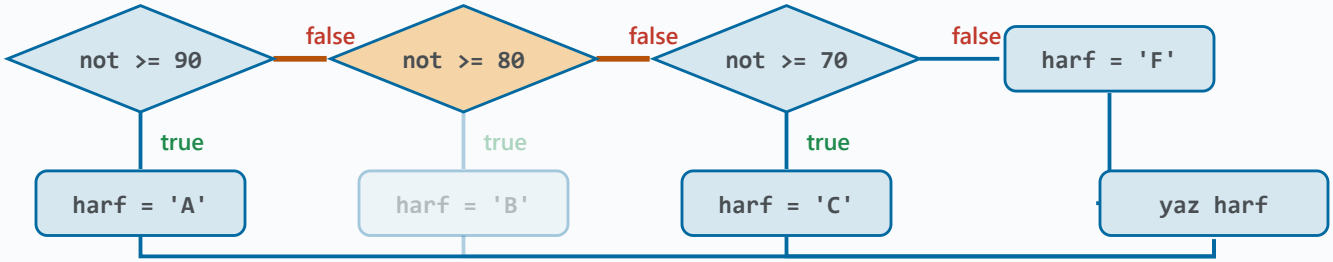
DEĞİŞKENLER

`not = 75` `harf = ?`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



2. koşul: not >= 80 → 75 >= 80 → **false**. Sonraki koşula geçilir.

Adım 3 / 16

```
1 int not = klavye.nextInt();
2 char harf;
3 if (not >= 90)      harf = 'A';
▶ else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

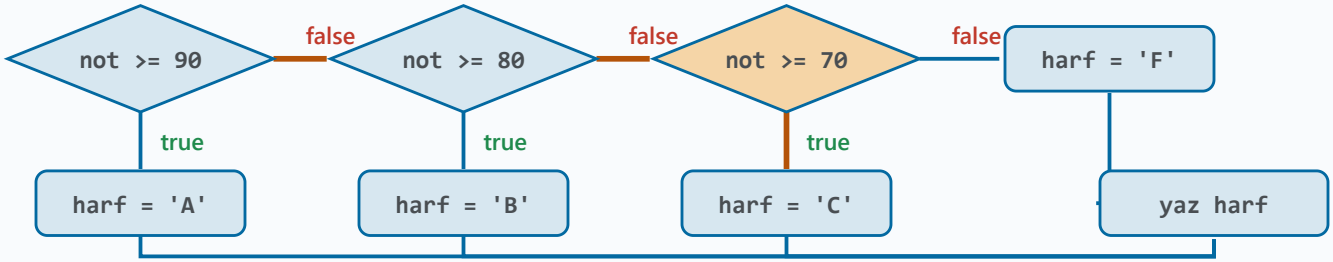
DEĞİŞKENLER

`not = 75` `harf = ?`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



3. koşul: not >= 70 → 75 >= 70 → **true**. Bu dal seçilir.

Adım 4 / 16

```
1 int not = klavye.nextInt();
2 char harf;
3 if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
▶ else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

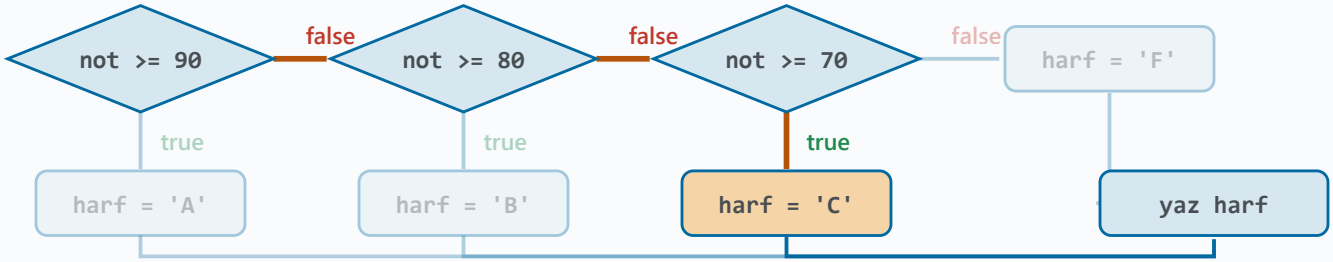
DEĞİŞKENLER

`not = 75` `harf = ?`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



harf = 'C'. Zincirin geri kalanı **denetlenmeden atlanır**.

Adım 5 / 16

```
1 int not = klavye.nextInt();
2 char harf;
3 if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
▶ else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

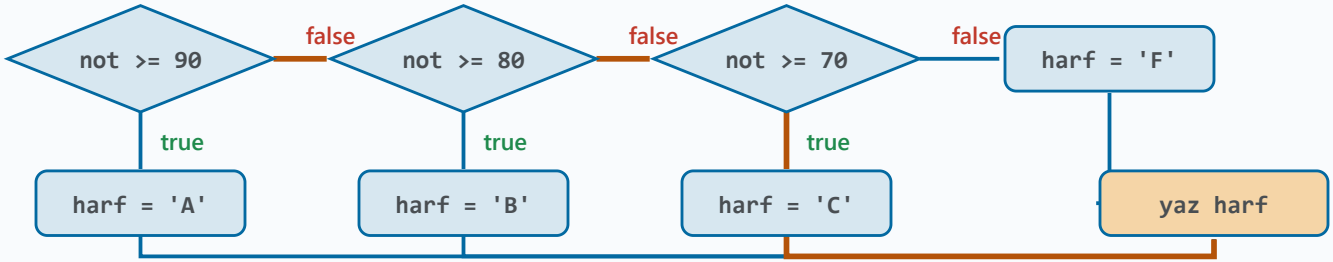
DEĞİŞKENLER

`not = 75` `harf = 'C'`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



C yazdırılır.

Adım 6 / 16

```
1 int not = klavye.nextInt();
2 char harf;
3 if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
▶ System.out.println(harf);
```

DEĞİŞKENLER

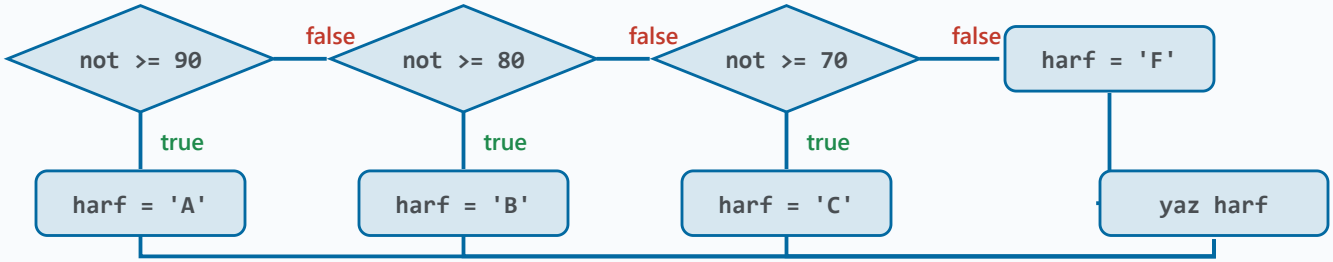
`not = 75` `harf = 'C'`

ÇIKTI

```
C
```

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



2. çalıştırma: not = **95**. harf bildirildi, henüz değeri yok.

Adım 7 / 16

```
▶ int not = klavye.nextInt();
▶ char harf;
3 if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

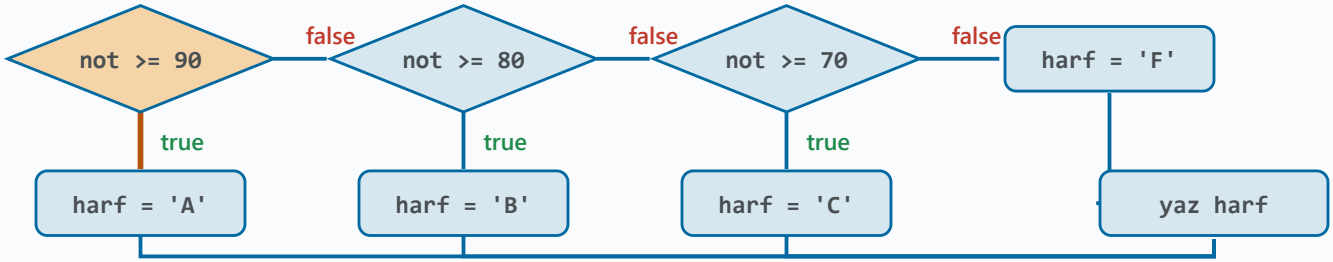
DEĞİŞKENLER

`not = 95` `harf = ?`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



1. koşul: not >= 90 → 95 >= 90 → **true**. Bu dal seçilir.

Adım 8 / 16

```
1 int not = klavye.nextInt();
2 char harf;
▶ if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

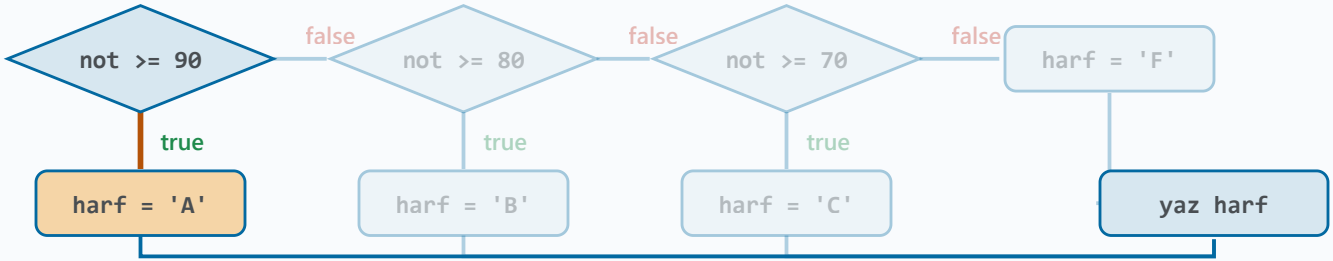
DEĞİŞKENLER

`not = 95` `harf = ?`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



harf = 'A'. Zincirin geri kalanı **denetlenmeden atlanır**.

Adım 9 / 16

```
1 int not = klavye.nextInt();
2 char harf;
▶ if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

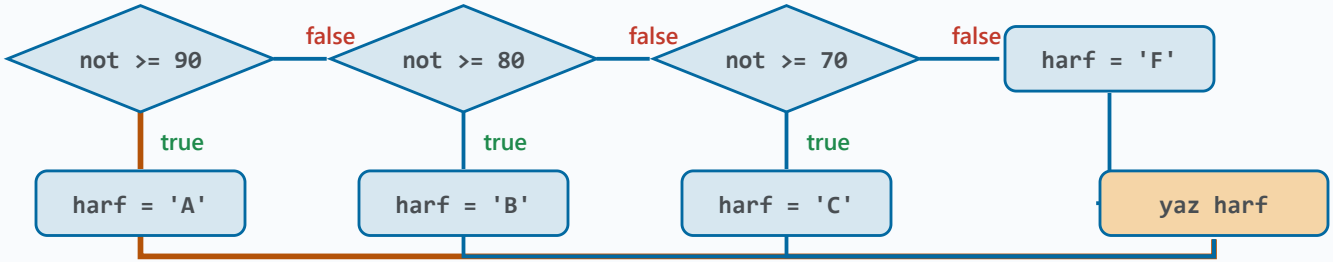
DEĞİŞKENLER

`not = 95` `harf = 'A'`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



A yazdırılır.

Adım 10 / 16

```
1 int not = klavye.nextInt();
2 char harf;
3 if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
▶ System.out.println(harf);
```

DEĞİŞKENLER

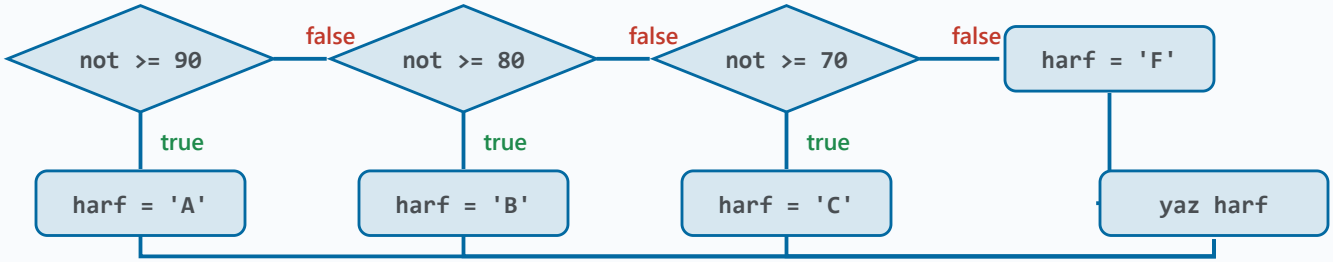
`not = 95` `harf = 'A'`

ÇIKTI

```
A
```

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



3. çalıştırma: not = **50**. harf bildirildi, henüz değeri yok.

Adım 11 / 16

```
▶ int not = klavye.nextInt();
▶ char harf;
3 if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

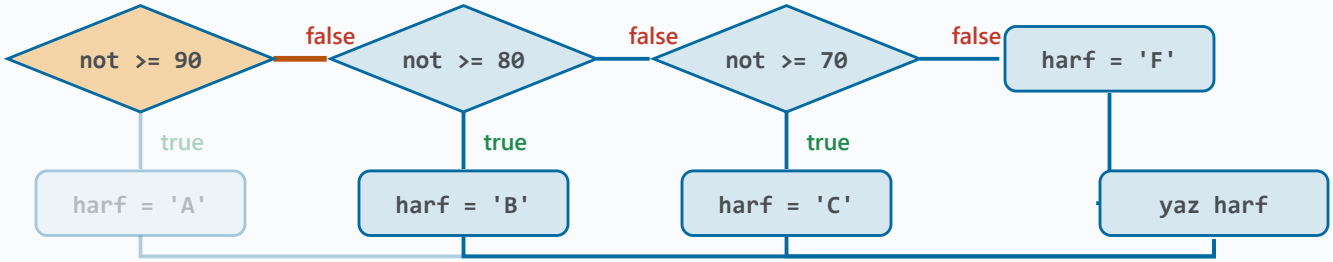
DEĞİŞKENLER

`not = 50` `harf = ?`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



1. koşul: not >= 90 → 50 >= 90 → **false**. Sonraki koşula geçilir.

Adım 12 / 16

```
1 int not = klavye.nextInt();
2 char harf;
▶ if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

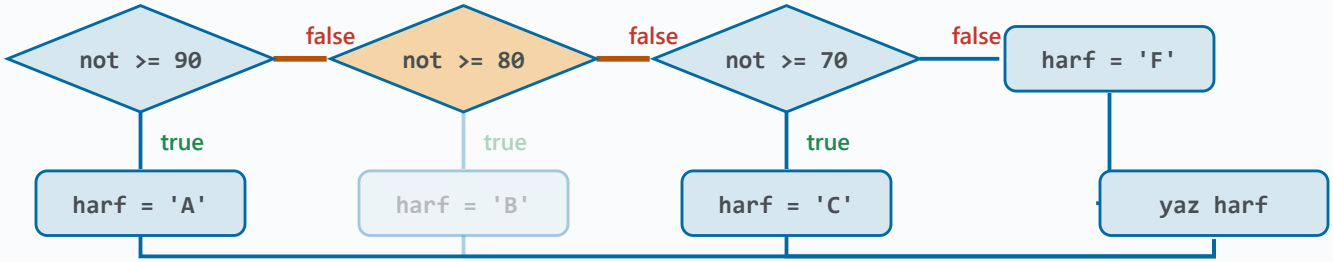
DEĞİŞKENLER

`not = 50` `harf = ?`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



2. koşul: not >= 80 → 50 >= 80 → **false**. Sonraki koşula geçilir.

Adım 13 / 16

```
1 int not = klavye.nextInt();
2 char harf;
3 if (not >= 90)      harf = 'A';
▶ else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

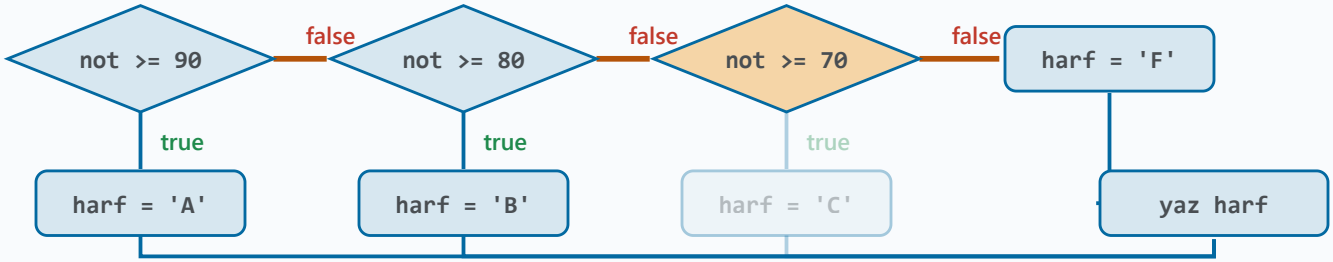
DEĞİŞKENLER

`not = 50` `harf = ?`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



3. koşul: not >= 70 → 50 >= 70 → **false**. Sonraki koşula geçilir.

Adım 14 / 16

```
1 int not = klavye.nextInt();
2 char harf;
3 if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
▶ else if (not >= 70) harf = 'C';
6 else                harf = 'F';
7 System.out.println(harf);
```

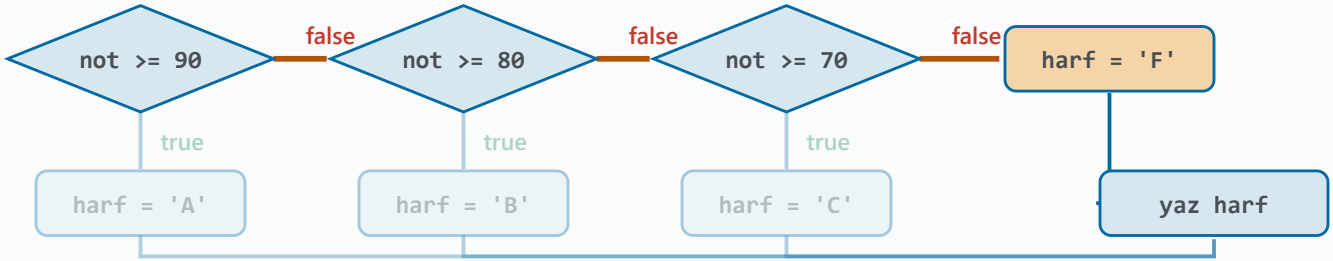
DEĞİŞKENLER

`not = 50` `harf = ?`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



Hiçbir koşul tutmadı → **else**: harf = 'F'.

Adım 15 / 16

```
1 int not = klavye.nextInt();
2 char harf;
3 if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
▶ else                harf = 'F';
7 System.out.println(harf);
```

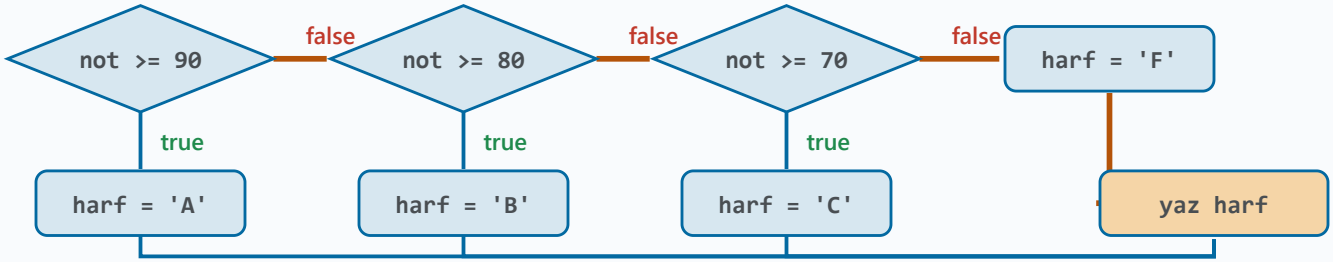
DEĞİŞKENLER

`not = 50` `harf = 'F'`

ÇIKTI

Tek deyimli dallarda süslü parantez yazılmayabilir.

else-if Zinciri: not = 75, 95, 50



F yazdırılır.

Adım 16 / 16

```
1 int not = klavye.nextInt();
2 char harf;
3 if (not >= 90)      harf = 'A';
4 else if (not >= 80) harf = 'B';
5 else if (not >= 70) harf = 'C';
6 else                harf = 'F';
▶ System.out.println(harf);
```

DEĞİŞKENLER

`not = 50` `harf = 'F'`

ÇIKTI

```
F
```

Tek deyimli dallarda süslü parantez yazılmayabilir.

Bağımsız if Zinciri: not = 95 → 3 satır

not = 95 atanır. Üç if de birbirinden bağımsızdır; hiçbiri diğerinin else dalı değildir.

Adım 1 / 4

```
▶ int not = 95;
2 if (not >= 90) System.out.println("A");
3 if (not >= 80) System.out.println("B");
4 if (not >= 70) System.out.println("C");
```

DEĞİŞKENLER

`not = 95`

ÇIKTI

Bağımsız if Zinciri: not = 95 → 3 satır

`95 >= 90` → true → **A** yazılır. Bağımsız if olduğu için sıradaki if **yine de** denenir.

Adım 2 / 4

```
1 int not = 95;
▶ if (not >= 90) System.out.println("A");
3 if (not >= 80) System.out.println("B");
4 if (not >= 70) System.out.println("C");
```

DEĞİŞKENLER

`not = 95`

ÇIKTI

```
A
```

Bağımsız if Zinciri: not = 95 → 3 satır

`95 >= 80` → true → **B** de yazılır. Önceki if'in sonucu bu koşulu etkilemez.

Adım 3 / 4

```
1 int not = 95;
2 if (not >= 90) System.out.println("A");
▶ if (not >= 80) System.out.println("B");
4 if (not >= 70) System.out.println("C");
```

DEĞİŞKENLER

`not = 95`

ÇIKTI

```
A
B
```

Bağımsız if Zinciri: not = 95 → 3 satır

`95 >= 70` → true → **C** de yazılır. Sonuç: 3 satır; tek harf beklenirken yanlış çıktı.

Adım 4 / 4

```
1 int not = 95;
2 if (not >= 90) System.out.println("A");
3 if (not >= 80) System.out.println("B");
▶ if (not >= 70) System.out.println("C");
```

DEĞİŞKENLER

`not = 95`

ÇIKTI

```
A
B
C
```

Aynı not = 95, else-if zinciri → 1 satır

Aynı **not = 95**; bu kez koşullar `else if` ile zincirlenmiş.

Adım 1 / 4

```
▶ int not = 95;
2 if (not >= 90) System.out.println("A");
3 else if (not >= 80) System.out.println("B");
4 else if (not >= 70) System.out.println("C");
```

DEĞİŞKENLER

`not = 95`

ÇIKTI

Aynı not = 95, else-if zinciri → 1 satır

`95 >= 90` → true → **A** yazılır.

Adım 2 / 4

```
1 int not = 95;
▶ if (not >= 90) System.out.println("A");
3 else if (not >= 80) System.out.println("B");
4 else if (not >= 70) System.out.println("C");
```

DEĞİŞKENLER

`not = 95`

ÇIKTI

```
A
```

Aynı not = 95, else-if zinciri → 1 satır

İlk dal çalıştığı için **else if** dalları hiç değerlendirilmez; 80 ve 70 koşulları denenmez.

Adım 3 / 4

```
1 int not = 95;
2 if (not >= 90) System.out.println("A");
▶ else if (not >= 80) System.out.println("B");
▶ else if (not >= 70) System.out.println("C");
```

DEĞİŞKENLER

`not = 95`

ÇIKTI

```
A
```

Aynı not = 95, else-if zinciri → 1 satır

Sonuç: yalnızca **1 satır**. Bağımsız if'te 3 satır (A, B, C), else-if zincirinde 1 satır (A) çıkar.

Adım 4 / 4

```
1 int not = 95;
2 if (not >= 90) System.out.println("A");
3 else if (not >= 80) System.out.println("B");
▶ else if (not >= 70) System.out.println("C");
```

DEĞİŞKENLER

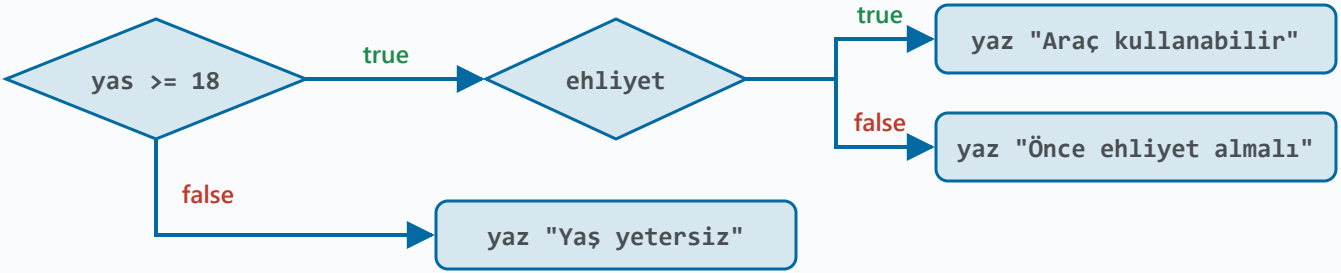
`not = 95`

ÇIKTI

```
A
```

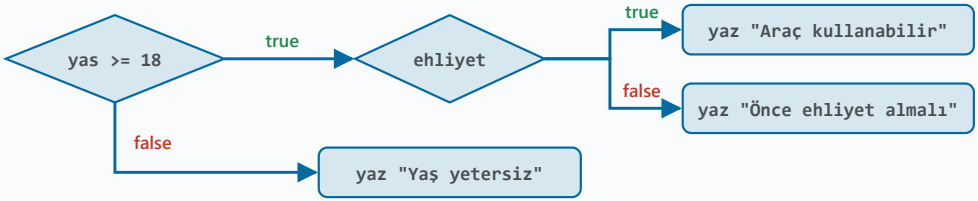
KARARLAR

İç İçe Kararlar



- Bir karar dalı içinde başka karar bulunabilir: iç koşul yalnızca dış koşul true olduğunda denetlenir.
- İç içe yapı, ikinci sorunun ancak ilk koşul sağlandığında anlamlı olduğu durumlar içindir.
- Derin iç içelik okunabilirliği azaltır; koşullar `&&` ile birleştirilebilir.

İç İçe Karar: 3 farklı girdi



1. çalıştırma: yas = 20, ehliyet = true.

Adım 1 / 11

```
▶ int yas = klavye.nextInt();
▶ boolean ehliyet = klavye.nextBoolean();
3 if (yas >= 18) {
4   if (ehliyet)
5     System.out.println("Araç kullanabilir");
6   else
7     System.out.println("Önce ehliyet almalı");
8 } else {
9   System.out.println("Yaş yetersiz");
10 }
```

DEĞİŞKENLER

`yas = 20` `ehliyet = true`

ÇIKTI

İç İçe Karar: 3 farklı girdi



Dış koşul: yas >= 18 → 20 >= 18 → **true**. İç if'e girilir.

Adım 2 / 11

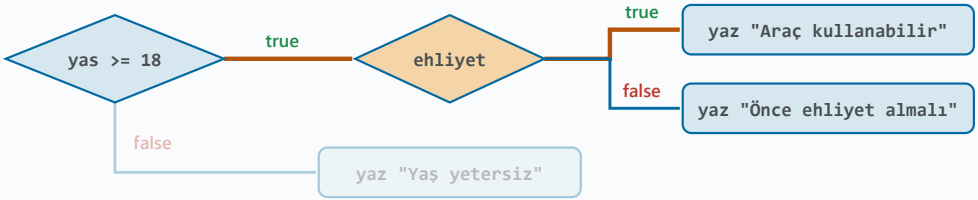
```
1  int yas = klavye.nextInt();
2  boolean ehliyet = klavye.nextBoolean();
▶ if (yas >= 18) {
4    if (ehliyet)
5      System.out.println("Araç kullanabilir");
6    else
7      System.out.println("Önce ehliyet almalı");
8  } else {
9    System.out.println("Yaş yetersiz");
10 }
```

DEĞİŞKENLER

`yas = 20` `ehliyet = true`

ÇIKTI

İç İçe Karar: 3 farklı girdi



İç koşul: ehliyet → **true**.

Adım 3 / 11

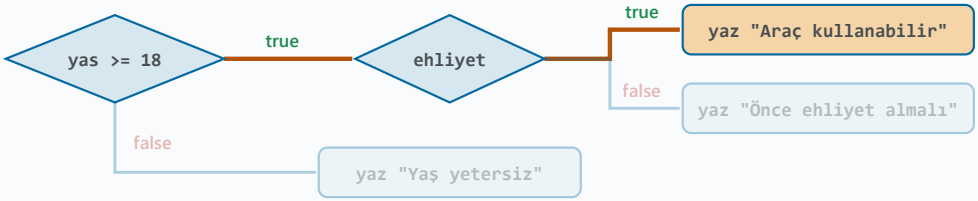
```
1  int yas = klavye.nextInt();
2  boolean ehliyet = klavye.nextBoolean();
3  if (yas >= 18) {
▶    if (ehliyet)
5      System.out.println("Araç kullanabilir");
6    else
7      System.out.println("Önce ehliyet almalı");
8  } else {
9    System.out.println("Yaş yetersiz");
10 }
```

DEĞİŞKENLER

`yas = 20` `ehliyet = true`

ÇIKTI

İç İçe Karar: 3 farklı girdi



Araç kullanabilir yazdırılır.

Adım 4 / 11

```
1  int yas = klavye.nextInt();
2  boolean ehliyet = klavye.nextBoolean();
3  if (yas >= 18) {
4    if (ehliyet)
▶      System.out.println("Araç kullanabilir");
6    else
7      System.out.println("Önce ehliyet almalı");
8  } else {
9    System.out.println("Yaş yetersiz");
10 }
```

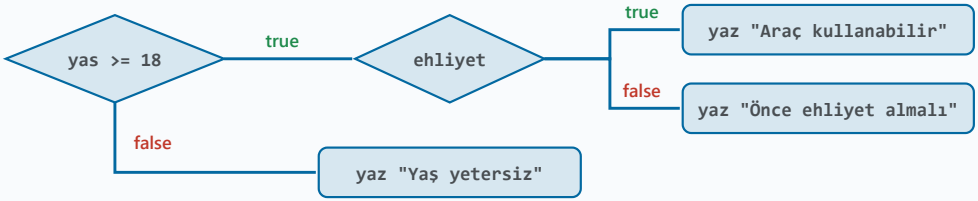
DEĞİŞKENLER

`yas = 20` `ehliyet = true`

ÇIKTI

```
Araç kullanabilir
```

İç İçe Karar: 3 farklı girdi



2. çalıştırma: yas = 20, ehliyet = false.

Adım 5 / 11

```
▶ int yas = klavye.nextInt();
▶ boolean ehliyet = klavye.nextBoolean();
3 if (yas >= 18) {
4   if (ehliyet)
5     System.out.println("Araç kullanabilir");
6   else
7     System.out.println("Önce ehliyet almalı");
8 } else {
9   System.out.println("Yaş yetersiz");
10 }
```

DEĞİŞKENLER

`yas = 20` `ehliyet = false`

ÇIKTI

İç İçe Karar: 3 farklı girdi



Dış koşul: yas >= 18 → 20 >= 18 → **true**. İç if'e girilir.

Adım 6 / 11

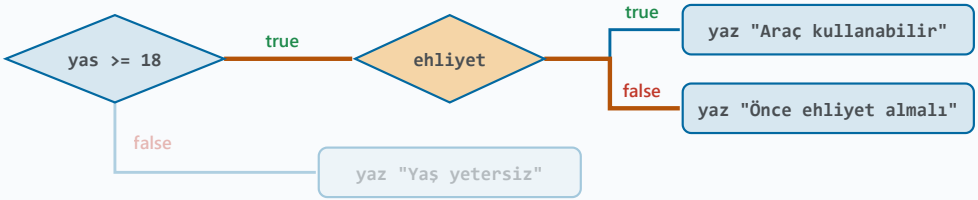
```
1  int yas = klavye.nextInt();
2  boolean ehliyet = klavye.nextBoolean();
▶ if (yas >= 18) {
4    if (ehliyet)
5      System.out.println("Araç kullanabilir");
6    else
7      System.out.println("Önce ehliyet almalı");
8  } else {
9    System.out.println("Yaş yetersiz");
10 }
```

DEĞİŞKENLER

`yas = 20` `ehliyet = false`

ÇIKTI

İç İçe Karar: 3 farklı girdi



İç koşul: ehliyet → **false**.

Adım 7 / 11

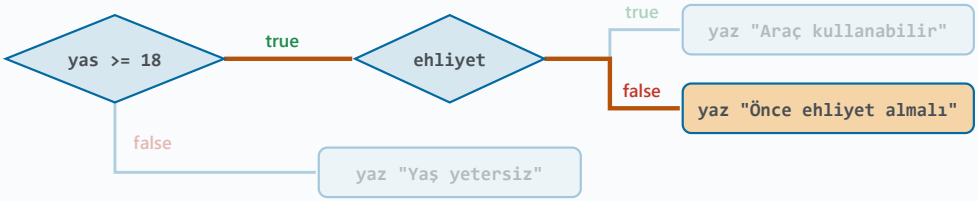
```
1  int yas = klavye.nextInt();
2  boolean ehliyet = klavye.nextBoolean();
3  if (yas >= 18) {
▶    if (ehliyet)
5      System.out.println("Araç kullanabilir");
6    else
7      System.out.println("Önce ehliyet almalı");
8  } else {
9    System.out.println("Yaş yetersiz");
10 }
```

DEĞİŞKENLER

`yas = 20` `ehliyet = false`

ÇIKTI

İç İçe Karar: 3 farklı girdi



Önce ehliyet almalı yazdırılır.

Adım 8 / 11

```
1  int yas = klavye.nextInt();
2  boolean ehliyet = klavye.nextBoolean();
3  if (yas >= 18) {
4    if (ehliyet)
5      System.out.println("Araç kullanabilir");
6    else
▶      System.out.println("Önce ehliyet almalı");
8  } else {
9    System.out.println("Yaş yetersiz");
10 }
```

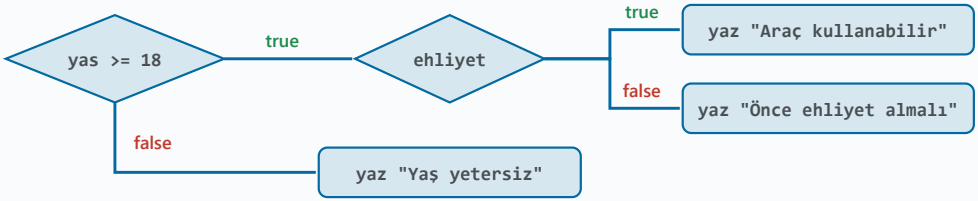
DEĞİŞKENLER

`yas = 20` `ehliyet = false`

ÇIKTI

```
Önce ehliyet almalı
```

İç İçe Karar: 3 farklı girdi



3. çalıştırma: yas = 15, ehliyet = true.

Adım 9 / 11

```
▶ int yas = klavye.nextInt();
▶ boolean ehliyet = klavye.nextBoolean();
3 if (yas >= 18) {
4   if (ehliyet)
5     System.out.println("Araç kullanabilir");
6   else
7     System.out.println("Önce ehliyet almalı");
8 } else {
9   System.out.println("Yaş yetersiz");
10 }
```

DEĞİŞKENLER

`yas = 15` `ehliyet = true`

ÇIKTI

İç İçe Karar: 3 farklı girdi



Dış koşul: 15 >= 18 → **false**. İç if **hiç denetlenmez**; ehliyet true olsa bile önemsizdir.

Adım 10 / 11

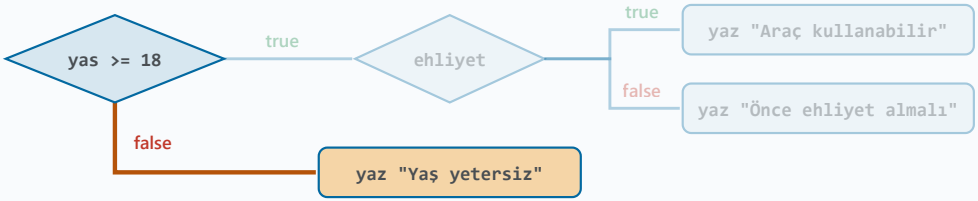
```
1  int yas = klavye.nextInt();
2  boolean ehliyet = klavye.nextBoolean();
▶  if (yas >= 18) {
4    if (ehliyet)
5      System.out.println("Araç kullanabilir");
6    else
7      System.out.println("Önce ehliyet almalı");
8  } else {
9    System.out.println("Yaş yetersiz");
10 }
```

DEĞİŞKENLER

`yas = 15` `ehliyet = true`

ÇIKTI

İç İçe Karar: 3 farklı girdi



Dış else çalışır: **Yaş yetersiz**.

Adım 11 / 11

```
1  int yas = klavye.nextInt();
2  boolean ehliyet = klavye.nextBoolean();
3  if (yas >= 18) {
4    if (ehliyet)
5      System.out.println("Araç kullanabilir");
6    else
7      System.out.println("Önce ehliyet almalı");
8  } else {
▶    System.out.println("Yaş yetersiz");
10 }
```

DEĞİŞKENLER

`yas = 15` `ehliyet = true`

ÇIKTI

```
Yaş yetersiz
```

Sarkan else (dangling else): a = −1

Başlangıç: **a = -1**, b = 5, x = 0. Girinti else'in dış if'e ait olduğunu düşündürür, ama derleyici girintiye bakmaz.

Adım 1 / 3

```
▶ int a = -1, b = 5, x = 0;
2 if (a > 0)
3     if (b > 0)
4         x = 1;
5     else
6         x = 2;
7 System.out.println(x);
```

DEĞİŞKENLER

a = **−1** b = **5** x = **0**

ÇIKTI

Sarkan else (dangling else): a = −1

`a > 0` → −1 > 0 → **false**. Dış if'in gövdesi (iç if ve onun else'i dahil) tümüyle atlanır.

Adım 2 / 3

```
1 int a = -1, b = 5, x = 0;
▶ if (a > 0)
3     if (b > 0)
4         x = 1;
5     else
6         x = 2;
7 System.out.println(x);
```

DEĞİŞKENLER

`a = -1` `b = 5` `x = 0`

ÇIKTI

Sarkan else (dangling else): a = −1

else en yakın if'e (`if (b > 0)`) bağlanır; o if hiç çalışmadığı için else de çalışmadı. x değişmedi: **0** yazılır.

Adım 3 / 3

```
1 int a = -1, b = 5, x = 0;
2 if (a > 0)
3     if (b > 0)
4         x = 1;
5     else
6         x = 2;
▶ System.out.println(x);
```

DEĞİŞKENLER

`a = −1` `b = 5` `x = 0`

ÇIKTI

```
0
```

Sarkan else'i Süslü Parantezle Düzeltmek

Aynı başlangıç değerleri. Bu kez iç if süslü parantezle kapatılmış; **else dış if'e** aittir.

Adım 1 / 4

```
▶ int a = -1, b = 5, x = 0;
2 if (a > 0) {
3     if (b > 0) x = 1;
4 } else {
5     x = 2;
6 }
7 System.out.println(x);
```

DEĞİŞKENLER

a = **-1** b = **5** x = **0**

ÇIKTI

Sarkan else'i Süslü Parantezle Düzeltmek

`a > 0` → **false**. İç if bloğu atlanır, `else` bloğuna geçilir.

Adım 2 / 4

```
1 int a = -1, b = 5, x = 0;
▶ if (a > 0) {
3     if (b > 0) x = 1;
▶ } else {
5     x = 2;
6 }
7 System.out.println(x);
```

DEĞİŞKENLER

`a = -1` `b = 5` `x = 0`

ÇIKTI

Sarkan else'i Süslü Parantezle Düzeltmek

else bloğu çalışır: **x = 2**.

Adım 3 / 4

```
1 int a = -1, b = 5, x = 0;
2 if (a > 0) {
3     if (b > 0) x = 1;
4 } else {
▶     x = 2;
6 }
7 System.out.println(x);
```

DEĞİŞKENLER

a = **−1** b = **5** x = **2**

ÇIKTI

Sarkan else'i Süslü Parantezle Düzeltmek

2 yazılır. Belirsizliği gidermek için iç içe if'lerde her zaman süslü parantez kullanın.

Adım 4 / 4

```
1 int a = -1, b = 5, x = 0;
2 if (a > 0) {
3     if (b > 0) x = 1;
4 } else {
5     x = 2;
6 }
▶ System.out.println(x);
```

DEĞİŞKENLER

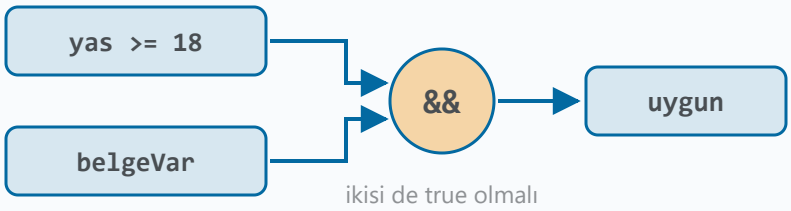
a = **−1** b = **5** x = **2**

ÇIKTI

```
2
```

Mantıksal İşleçlerle Karar

```
boolean uygun = yas >= 18 && belgeVar;
if (uygun) {
    System.out.println("Başvuru alınabilir");
}
```



yas >= 18 ↓ \ belgeVar	true	false
true	true	false
false	false	false

uygun değeri

- && iki koşulun da doğru olmasını ister.
- || seçeneklerden en az birinin doğru olmasını ister.

&& ile Karar: 3 farklı girdi



1. çalıştırma: yas = 20, belgeVar = true.

Adım 1 / 9

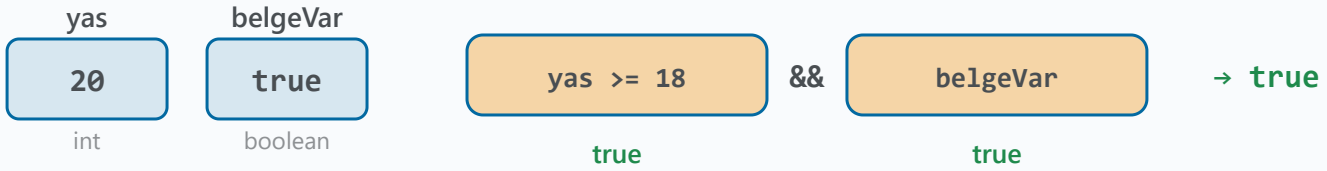
```
▶ int yas = klavye.nextInt();
▶ boolean belgeVar = klavye.nextBoolean();
3 boolean uygun = yas >= 18 && belgeVar;
4 if (uygun) {
5     System.out.println("Başvuru alınabilir");
6 }
```

DEĞİŞKENLER

yas = 20 | belgeVar = true

ÇIKTI

&& ile Karar: 3 farklı girdi



20 >= 18 → true; sağ taraf gerekli: belgeVar → true. true && true → **true**.

Adım 2 / 9

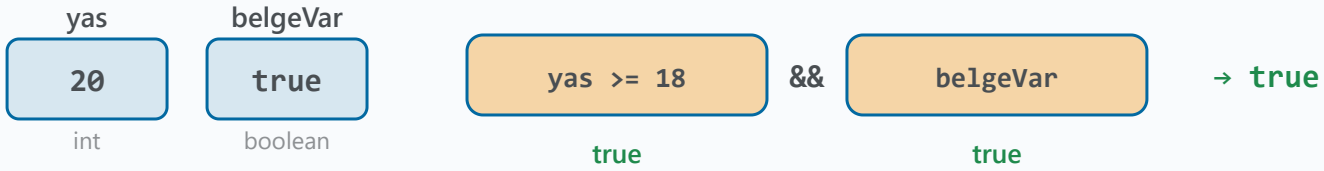
```
1 int yas = klavye.nextInt();
2 boolean belgeVar = klavye.nextBoolean();
▶ boolean uygun = yas >= 18 && belgeVar;
4 if (uygun) {
5     System.out.println("Başvuru alınabilir");
6 }
```

DEĞİŞKENLER

`yas = 20` `belgeVar = true` `uygun = true`

ÇIKTI

`&&` ile Karar: 3 farklı girdi



Başvuru alınabilir yazdırılır.

Adım 3 / 9

```
1 int yas = klavye.nextInt();
2 boolean belgeVar = klavye.nextBoolean();
3 boolean uygun = yas >= 18 && belgeVar;
4 if (uygun) {
▶     System.out.println("Başvuru alınabilir");
6 }
```

DEĞİŞKENLER

`yas = 20` `belgeVar = true` `uygun = true`

ÇIKTI

```
Başvuru alınabilir
```

&& ile Karar: 3 farklı girdi



2. çalıştırma: yas = 20, belgeVar = false.

Adım 4 / 9

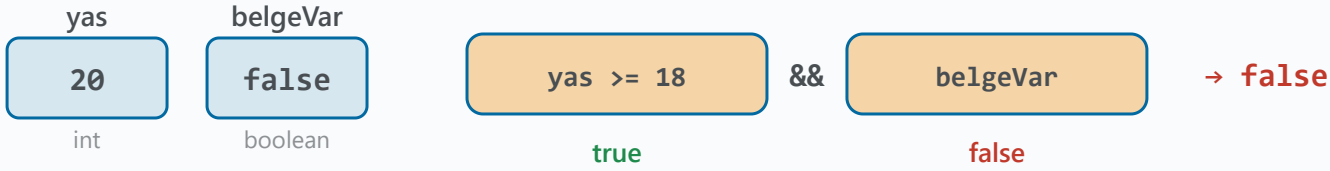
```
▶ int yas = klavye.nextInt();
▶ boolean belgeVar = klavye.nextBoolean();
3 boolean uygun = yas >= 18 && belgeVar;
4 if (uygun) {
5     System.out.println("Başvuru alınabilir");
6 }
```

DEĞİŞKENLER

yas = 20 belgeVar = false

ÇIKTI

&& ile Karar: 3 farklı girdi



20 >= 18 → true; sağ taraf gerekli: belgeVar → false. true && false → **false**.

Adım 5 / 9

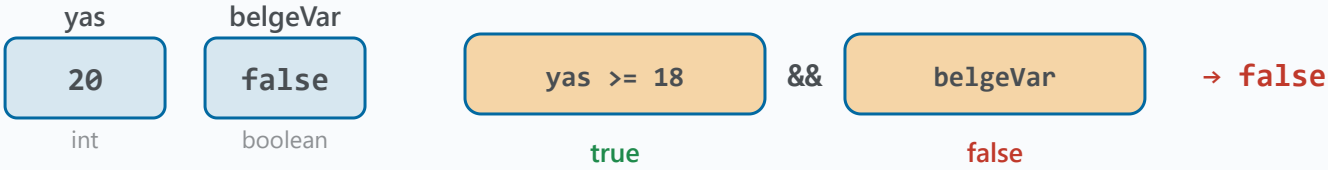
```
1 int yas = klavye.nextInt();
2 boolean belgeVar = klavye.nextBoolean();
▶ boolean uygun = yas >= 18 && belgeVar;
4 if (uygun) {
5     System.out.println("Başvuru alınabilir");
6 }
```

DEĞİŞKENLER

yas = 20 | belgeVar = false | uygun = false

ÇIKTI

&& ile Karar: 3 farklı girdi



uygun false → blok atlanır, çıktı yok.

Adım 6 / 9

```
1 int yas = klavye.nextInt();
2 boolean belgeVar = klavye.nextBoolean();
3 boolean uygun = yas >= 18 && belgeVar;
▶ if (uygun) {
5     System.out.println("Başvuru alınabilir");
6 }
```

DEĞİŞKENLER

`yas = 20` `belgeVar = false` `uygun = false`

ÇIKTI

&& ile Karar: 3 farklı girdi



3. çalıştırma: yas = 16, belgeVar = true.

Adım 7 / 9

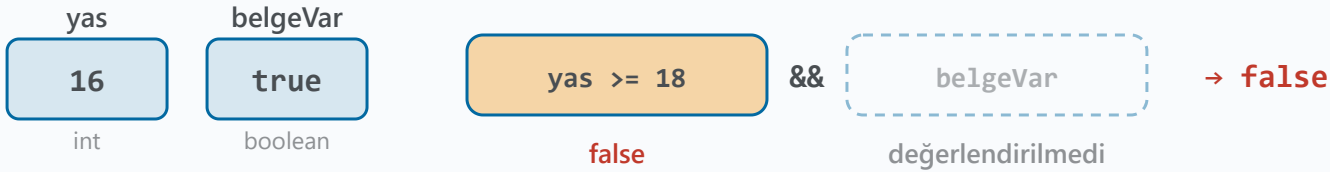
```
▶ int yas = klavye.nextInt();
▶ boolean belgeVar = klavye.nextBoolean();
3 boolean uygun = yas >= 18 && belgeVar;
4 if (uygun) {
5     System.out.println("Başvuru alınabilir");
6 }
```

DEĞİŞKENLER

yas = 16 belgeVar = true

ÇIKTI

&& ile Karar: 3 farklı girdi



16 >= 18 → **false**. Kısa devre: belgeVar **okunmaz**, sonuç kesin false.

Adım 8 / 9

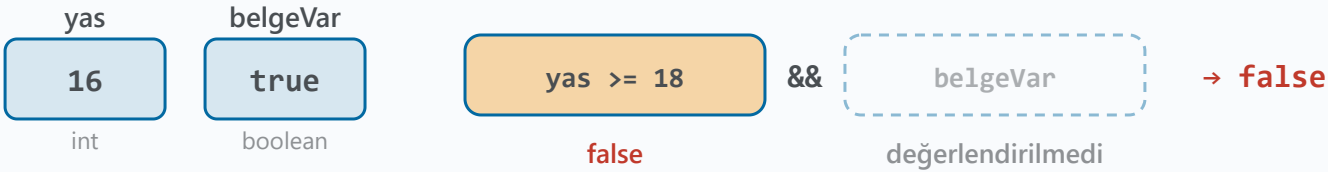
```
1 int yas = klavye.nextInt();
2 boolean belgeVar = klavye.nextBoolean();
▶ boolean uygun = yas >= 18 && belgeVar;
4 if (uygun) {
5     System.out.println("Başvuru alınabilir");
6 }
```

DEĞİŞKENLER

yas = 16 | belgeVar = true | uygun = false

ÇIKTI

&& ile Karar: 3 farklı girdi



uygun false → blok atlanır, çıktı yok.

Adım 9 / 9

```
1 int yas = klavye.nextInt();
2 boolean belgeVar = klavye.nextBoolean();
3 boolean uygun = yas >= 18 && belgeVar;
▶ if (uygun) {
5     System.out.println("Başvuru alınabilir");
6 }
```

DEĞİŞKENLER

`yas = 16` `belgeVar = true` `uygun = false`

ÇIKTI

! ve De Morgan Kuralları

a	b	!(a && b)	!a || !b	!(a || b)	!a && !b
false	false	true	true	true	true
false	true	true	true	false	false
true	false	true	true	false	false
true	true	false	false	false	false

```
// Başvuru reddedilsin: 18 yaşından küçük ya da belge yok
if (!(yas >= 18 && belgeVar)) { … }
// De Morgan ile aynı koşul:
if (yas < 18 || !belgeVar) { … }
```

- `!` parantezin içine dağıtılırken `&&` ile `||` yer değiştirir, her terim de tersine döner.
- Karşılaştırmaların tersi: `>=` ↔ `<`, `>` ↔ `<=`, `==` ↔ `!=`.

KARARLAR

Kısa Devresiz `&` ve `|`

```
int a = 10, b = 0;
if (b != 0 && a / b > 1) { … }  // güvenli: a / b hiç hesaplanmaz
if (b != 0 &  a / b > 1) { … }  // ArithmeticException: / by zero
```

Tek karakterli `&` ve `|` boolean değerlerde de kullanılır, ama **iki tarafı da her zaman** değerlendirir.

```
static int sayac = 0;
static boolean kontrol(boolean v) { sayac++; return v; }
```

İfade (sayac 0'dan başlar)	sayac
`kontrol(false) && kontrol(true)`	1
`kontrol(false) & kontrol(true)`	2
`kontrol(true) || kontrol(true)`	1
`kontrol(true) | kontrol(true)`	2

Koşul İşleci: `(a > b) ? a : b`



a = 7, b = 12.

Adım 1 / 4

```
▶ int a = 7, b = 12;
2 int enBuyuk = (a > b) ? a : b;
3 System.out.println(enBuyuk);
```

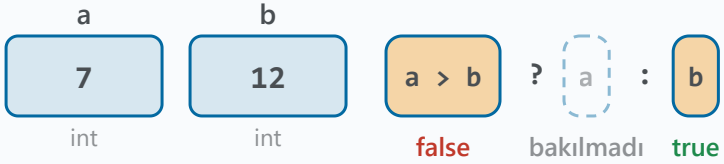
DEĞİŞKENLER

`a = 7` `b = 12`

ÇIKTI

Kısa, iki seçenekli değer seçimlerinde kullanın; iç içe ?: okunabilirliği bozar.

Koşul İşleci: `(a > b) ? a : b`



Koşul: a > b → 7 > 12 → **false**. false olduğu için **:**'dan sonraki değer seçilir.

Adım 2 / 4

```
1 int a = 7, b = 12;
▶ int enBuyuk = (a > b) ? a : b;
3 System.out.println(enBuyuk);
```

DEĞİŞKENLER

`a = 7` `b = 12`

ÇIKTI

Kısa, iki seçenekli değer seçimlerinde kullanın; iç içe ?: okunabilirliği bozar.

Koşul İşleci: `(a > b) ? a : b`



enBuyuk = 12. `?:` bir **ifadedir**: değer üretir, atamada kullanılabilir.

Adım 3 / 4

```
1 int a = 7, b = 12;
▶ int enBuyuk = (a > b) ? a : b;
3 System.out.println(enBuyuk);
```

DEĞİŞKENLER

`a = 7` `b = 12` `enBuyuk = 12`

ÇIKTI

Kısa, iki seçenekli değer seçimlerinde kullanın; iç içe ?: okunabilirliği bozar.

Koşul İşleci: `(a > b) ? a : b`



12 yazdırılır. Aynı işi if-else 4 satırda yapar.

Adım 4 / 4

```
1 int a = 7, b = 12;
2 int enBuyuk = (a > b) ? a : b;
▶ System.out.println(enBuyuk);
```

DEĞİŞKENLER

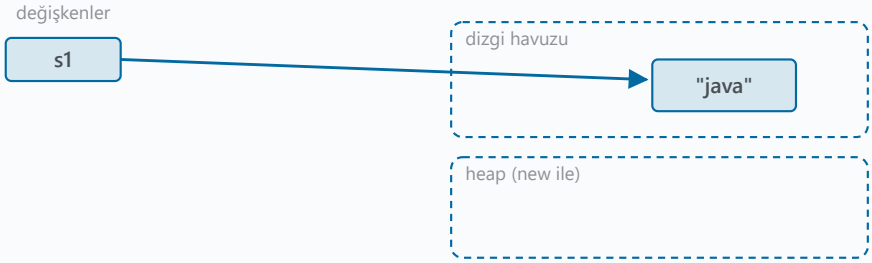
`a = 7` `b = 12` `enBuyuk = 12`

ÇIKTI

```
12
```

Kısa, iki seçenekli değer seçimlerinde kullanın; iç içe ?: okunabilirliği bozar.

String Karşılaştırma: `==` ile `equals`



Sabit `"java"` dizgi havuzunda oluşturulur; **s1** bu nesneye referans tutar.

Adım 1 / 6

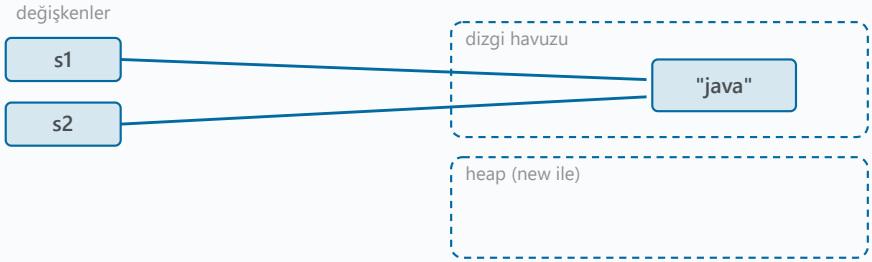
```
1 String s1 = "java";
2 String s2 = "java";
3 String s3 = new String("java");
4 System.out.println(s1 == s2);
5 System.out.println(s1 == s3);
6 System.out.println(s1.equals(s3));
```

DEĞİŞKENLER

`s1 = →havuz`

ÇIKTI

String Karşılaştırma: `==` ile `equals`



Aynı sabit yeniden yazıldığında havuzdaki nesne **paylaşılır**; s2 de aynı nesneyi gösterir.

Adım 2 / 6

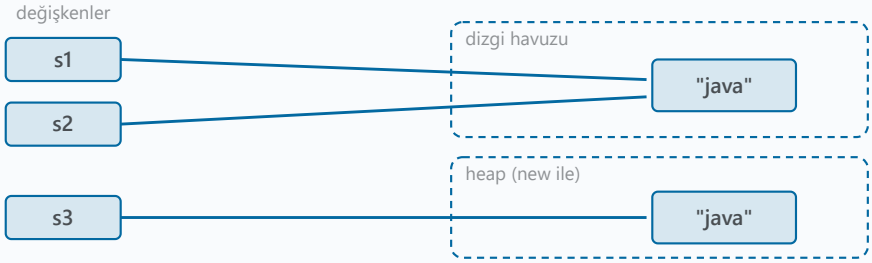
```
1 String s1 = "java";
▶ String s2 = "java";
3 String s3 = new String("java");
4 System.out.println(s1 == s2);
5 System.out.println(s1 == s3);
6 System.out.println(s1.equals(s3));
```

DEĞİŞKENLER

`s1 = →havuz` `s2 = →havuz`

ÇIKTI

String Karşılaştırma: `==` ile `equals`



`new String("java")` her zaman **yeni bir nesne** üretir; s3 heap'teki ayrı nesneyi gösterir.

Adım 3 / 6

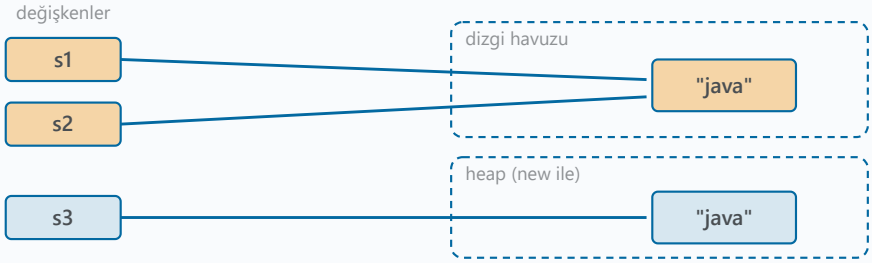
```
1 String s1 = "java";
2 String s2 = "java";
▶ String s3 = new String("java");
4 System.out.println(s1 == s2);
5 System.out.println(s1 == s3);
6 System.out.println(s1.equals(s3));
```

DEĞİŞKENLER

`s1 = →havuz` `s2 = →havuz` `s3 = →heap`

ÇIKTI

String Karşılaştırma: `==` ile `equals`



`==` referansları (adresleri) karşılaştırır. s1 ve s2 aynı nesneyi gösterdiği için **true**.

Adım 4 / 6

```
1 String s1 = "java";
2 String s2 = "java";
3 String s3 = new String("java");
▶ System.out.println(s1 == s2);
5 System.out.println(s1 == s3);
6 System.out.println(s1.equals(s3));
```

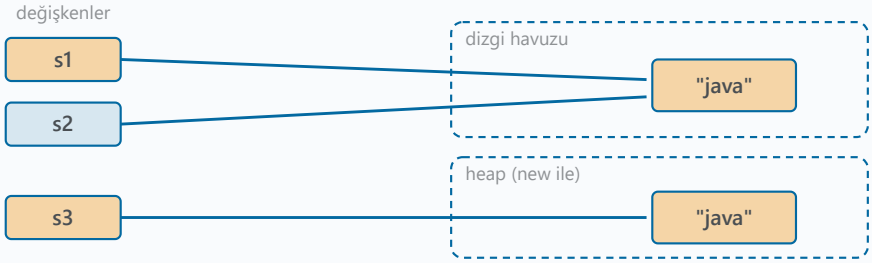
DEĞİŞKENLER

`s1 = →havuz` `s2 = →havuz` `s3 = →heap`

ÇIKTI

```
true
```

String Karşılaştırma: `==` ile `equals`



s1 ve s3 içerik olarak aynı ama nesneler farklı → adresler farklı → **false**.

Adım 5 / 6

```
1 String s1 = "java";
2 String s2 = "java";
3 String s3 = new String("java");
4 System.out.println(s1 == s2);
▶ System.out.println(s1 == s3);
6 System.out.println(s1.equals(s3));
```

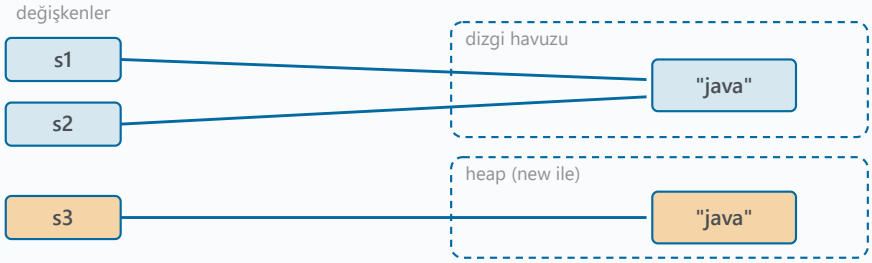
DEĞİŞKENLER

`s1 = →havuz` `s2 = →havuz` `s3 = →heap`

ÇIKTI

```
true
false
```

String Karşılaştırma: `==` ile `equals`



`equals` nesnelerin **içeriğini** karşılaştırır: "java" = "java" → **true**. Dizgileri hep equals ile karşılaştırın.

Adım 6 / 6

```
1 String s1 = "java";
2 String s2 = "java";
3 String s3 = new String("java");
4 System.out.println(s1 == s2);
5 System.out.println(s1 == s3);
▶ System.out.println(s1.equals(s3));
```

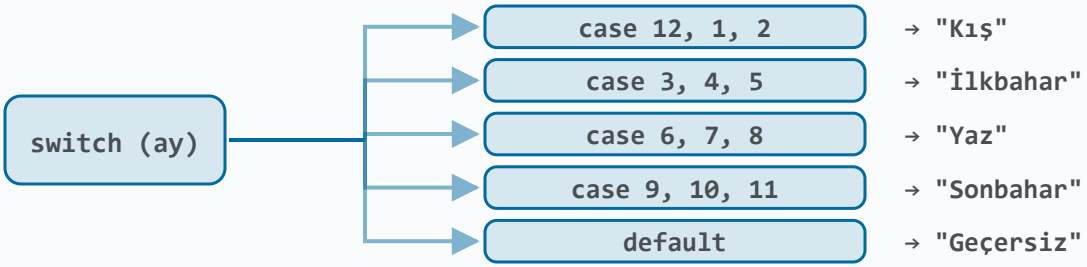
DEĞİŞKENLER

`s1 = →havuz` `s2 = →havuz` `s3 = →heap`

ÇIKTI

```
true
false
true
```

switch Ne Zaman?

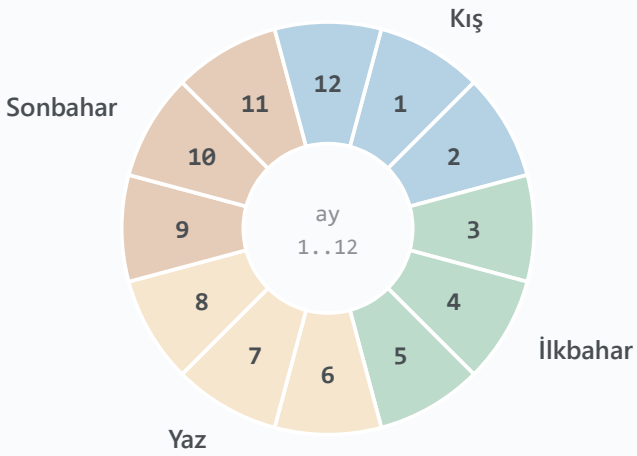


- Tek bir ifadenin **ayrık değerlerine** göre seçim yaparken okunabilirdir.
- case etiketleri sabit seçenekleri temsil eder; eşleşen case'e doğrudan gidilir.
- Aralık ve karmaşık mantıksal koşullarda if-else çoğunlukla daha uygundur.

SWITCH

switch İfadesi

```
String mevsim = switch (ay) {
    case 12, 1, 2 -> "Kış";
    case 3, 4, 5 -> "İlkbahar";
    case 6, 7, 8 -> "Yaz";
    case 9, 10, 11 -> "Sonbahar";
    default -> "Geçersiz";
};
```

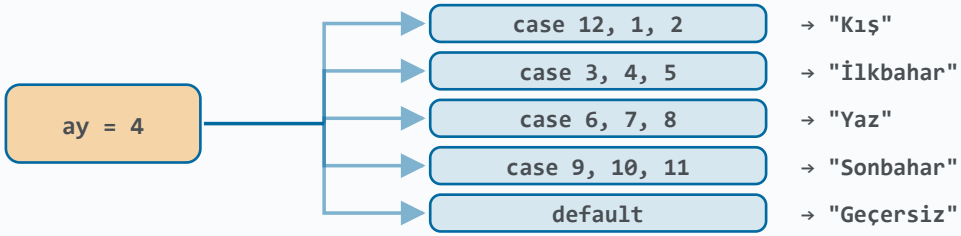


13 gibi değer → default: "Geçersiz"

- Ok biçimi yanlışlıkla sonraki case'e geçişi önler.
- default beklenmeyen değerleri ele alır. Switch **ifadesinde** (`x = switch …`) tüm değerler kapsanmalıdır; int ve String için `default` zorunludur, switch deyiminde isteğe bağlıdır. Bu yazım Java 14 ve sonrası içindir.

SWITCH · KODU ADIM ADIM ÇALIŞTIR

switch İfadesi: ay = 4, 12, 13



1. çalıştırma: ay = **4**.

Adım 1 / 12

```
▶ int ay = klavye.nextInt();
2 String mevsim = switch (ay) {
3     case 12, 1, 2 -> "Kış";
4     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
9 System.out.println(mevsim);
```

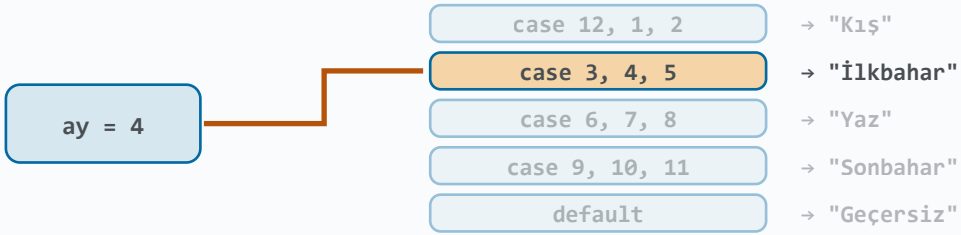
DEĞİŞKENLER

ay = 4

ÇIKTI

SWITCH · KODU ADIM ADIM ÇALIŞTIR

switch İfadesi: ay = 4, 12, 13



switch (ay) → 4: 4 bu case'in listesinde. Seçilen dalın değeri: **"İlkbahar"**.

Adım 2 / 12

```
1 int ay = klavye.nextInt();
2 String mevsim = switch (ay) {
3     case 12, 1, 2 -> "Kış";
▶     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
9 System.out.println(mevsim);
```

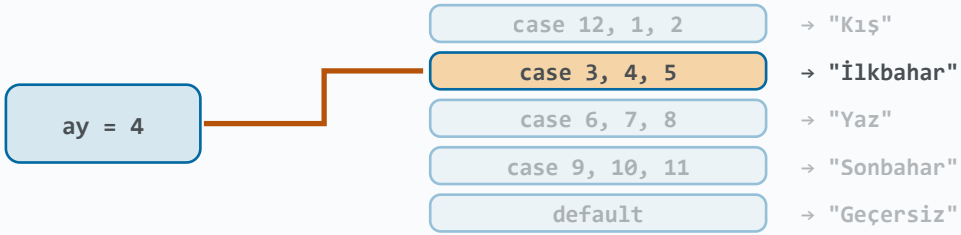
DEĞİŞKENLER

ay = 4

ÇIKTI

SWITCH · KODU ADIM ADIM ÇALIŞTIR

switch İfadesi: ay = 4, 12, 13



switch ifadesinin ürettiği değer mevsim'e atanır: mevsim = "İlkbahar". Başka case çalışmaz.

Adım 3 / 12

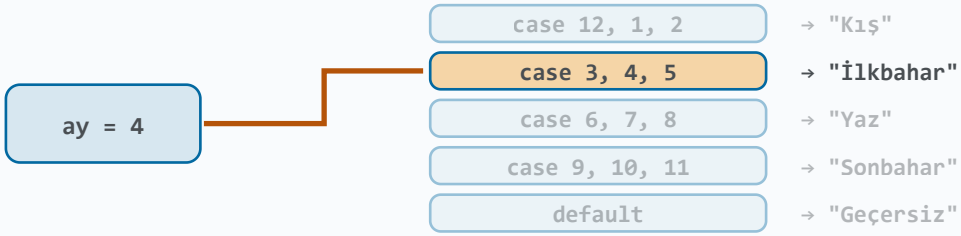
```
1 int ay = klavye.nextInt();
▶ String mevsim = switch (ay) {
3     case 12, 1, 2 -> "Kış";
4     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
9 System.out.println(mevsim);
```

DEĞİŞKENLER

`ay = 4` `mevsim = "İlkbahar"`

ÇIKTI

switch İfadesi: ay = 4, 12, 13



İlkbahar yazdırılır.

Adım 4 / 12

```
1 int ay = klavye.nextInt();
2 String mevsim = switch (ay) {
3     case 12, 1, 2 -> "Kış";
4     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
▶ System.out.println(mevsim);
```

DEĞİŞKENLER

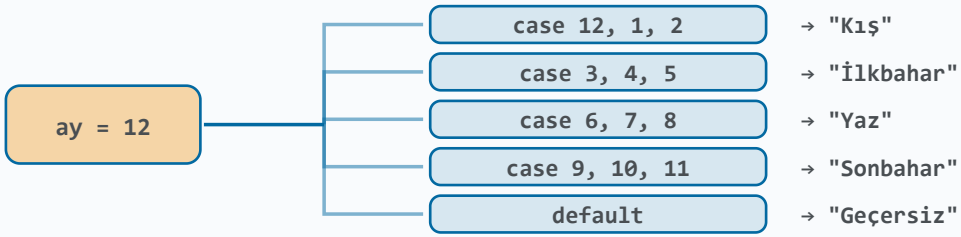
`ay = 4` `mevsim = "İlkbahar"`

ÇIKTI

```
İlkbahar
```

SWITCH · KODU ADIM ADIM ÇALIŞTIR

switch İfadesi: ay = 4, 12, 13



2. çalıştırma: ay = **12**.

Adım 5 / 12

```
▶ int ay = klavye.nextInt();
2 String mevsim = switch (ay) {
3     case 12, 1, 2 -> "Kış";
4     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
9 System.out.println(mevsim);
```

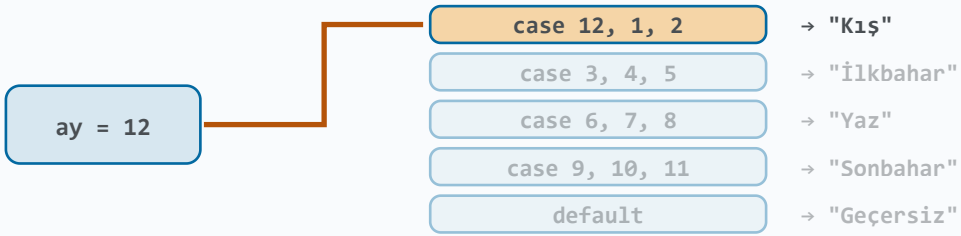
DEĞİŞKENLER

ay = 12

ÇIKTI

SWITCH · KODU ADIM ADIM ÇALIŞTIR

switch İfadesi: ay = 4, 12, 13



switch (ay) → 12: 12 bu case'in listesinde. Seçilen dalın değeri: **"Kış"**.

Adım 6 / 12

```
1 int ay = klavye.nextInt();
2 String mevsim = switch (ay) {
▶     case 12, 1, 2 -> "Kış";
4     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
9 System.out.println(mevsim);
```

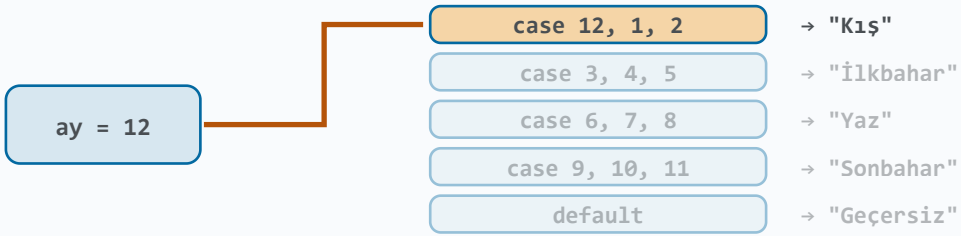
DEĞİŞKENLER

ay = **12**

ÇIKTI

SWITCH · KODU ADIM ADIM ÇALIŞTIR

switch İfadesi: ay = 4, 12, 13



switch ifadesinin ürettiği değer mevsim'e atanır: mevsim = "Kış". Başka case çalışmaz.

Adım 7 / 12

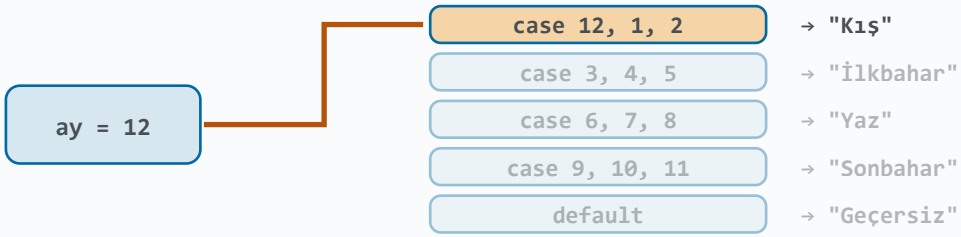
```
1 int ay = klavye.nextInt();
▶ String mevsim = switch (ay) {
3     case 12, 1, 2 -> "Kış";
4     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
9 System.out.println(mevsim);
```

DEĞİŞKENLER

`ay = 12` `mevsim = "Kış"`

ÇIKTI

switch İfadesi: ay = 4, 12, 13



Kış yazdırılır.

Adım 8 / 12

```
1 int ay = klavye.nextInt();
2 String mevsim = switch (ay) {
3     case 12, 1, 2 -> "Kış";
4     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
▶ System.out.println(mevsim);
```

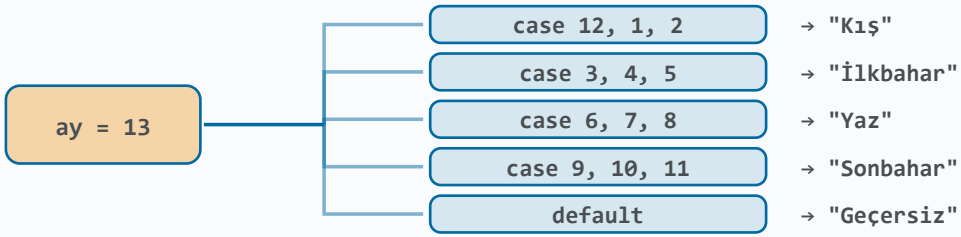
DEĞİŞKENLER

`ay = 12` `mevsim = "Kış"`

ÇIKTI

```
Kış
```

switch İfadesi: ay = 4, 12, 13



3. çalıştırma: ay = **13**.

Adım 9 / 12

```
▶ int ay = klavye.nextInt();
2 String mevsim = switch (ay) {
3     case 12, 1, 2 -> "Kış";
4     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
9 System.out.println(mevsim);
```

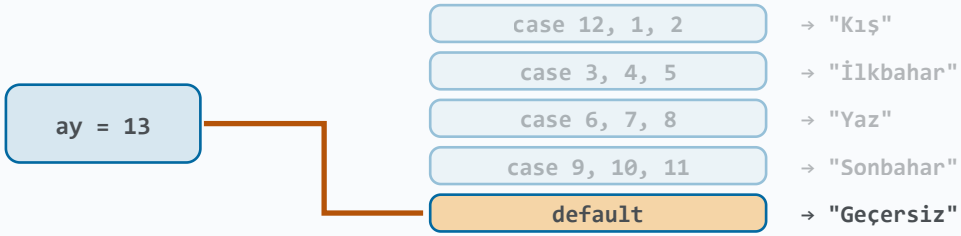
DEĞİŞKENLER

ay = 13

ÇIKTI

SWITCH · KODU ADIM ADIM ÇALIŞTIR

switch İfadesi: ay = 4, 12, 13



switch (ay) → 13: 13 hiçbir case'te yok → **default**. Seçilen dalın değeri: **"Geçersiz"**.

Adım 10 / 12

```
1  int ay = klavye.nextInt();
2  String mevsim = switch (ay) {
3      case 12, 1, 2 -> "Kış";
4      case 3, 4, 5 -> "İlkbahar";
5      case 6, 7, 8 -> "Yaz";
6      case 9, 10, 11 -> "Sonbahar";
▶      default -> "Geçersiz";
8  };
9  System.out.println(mevsim);
```

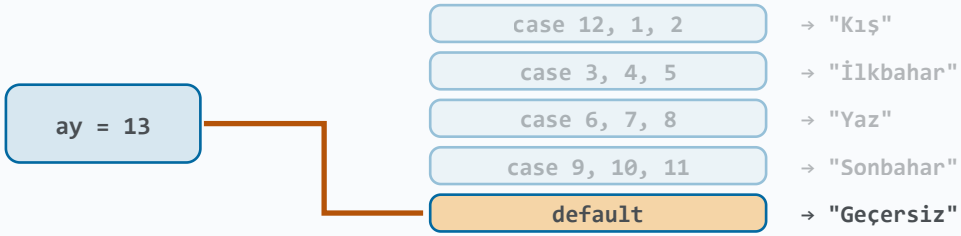
DEĞİŞKENLER

`ay = 13`

ÇIKTI

SWITCH · KODU ADIM ADIM ÇALIŞTIR

switch İfadesi: ay = 4, 12, 13



switch ifadesinin ürettiği değer mevsim'e atanır: mevsim = "Geçersiz". Başka case çalışmaz.

Adım 11 / 12

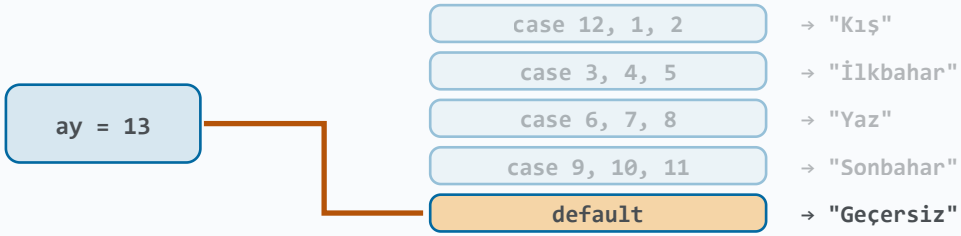
```
1 int ay = klavye.nextInt();
▶ String mevsim = switch (ay) {
3     case 12, 1, 2 -> "Kış";
4     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
9 System.out.println(mevsim);
```

DEĞİŞKENLER

`ay = 13` `mevsim = "Geçersiz"`

ÇIKTI

switch İfadesi: ay = 4, 12, 13



Geçersiz yazdırılır.

Adım 12 / 12

```
1 int ay = klavye.nextInt();
2 String mevsim = switch (ay) {
3     case 12, 1, 2 -> "Kış";
4     case 3, 4, 5 -> "İlkbahar";
5     case 6, 7, 8 -> "Yaz";
6     case 9, 10, 11 -> "Sonbahar";
7     default -> "Geçersiz";
8 };
▶ System.out.println(mevsim);
```

DEĞİŞKENLER

`ay = 13` `mevsim = "Geçersiz"`

ÇIKTI

```
Geçersiz
```

SWITCH

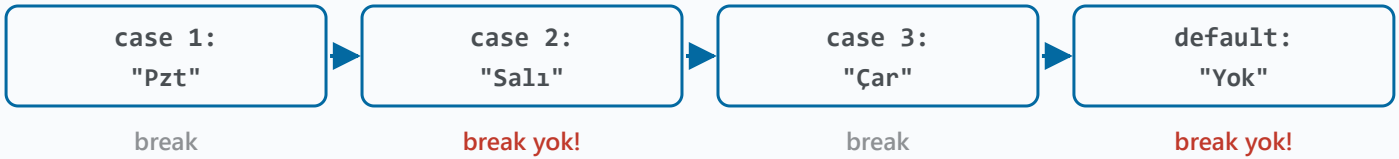
switch İfadesinde `yield`

```
int yil = 2024, ay = 2;
int gun = switch (ay) {
    case 4, 6, 9, 11 -> 30;
    case 2 -> {
        System.out.println("Şubat");
        yield yil % 4 == 0 ? 29 : 28;
    }
    default -> 31;
};
System.out.println(gun);
```

Çıktı: `Şubat` ve `29` (iki satır).

- Tek ifadeli dalda değer doğrudan yazılır; birden çok deyim gerekiyorsa dal `{ }` bloğuna alınır ve değer **`yield`** ile verilir.
- Blok `yield` olmadan biterse derleme hatası olur: `switch rule completes without providing a value`.
- Java 14 ve sonrası içindir; artık yıl denetimi burada basitleştirilmiştir (100/400 kuralı yok).

Klasik switch: `break` Unutulursa



gun = 2.

Adım 1 / 5

```
▶ int gun = 2;
2 switch (gun) {
3   case 1: System.out.println("Pzt"); break;
4   case 2: System.out.println("Salı");
5   case 3: System.out.println("Çar"); break;
6   default: System.out.println("Yok");
7 }
```

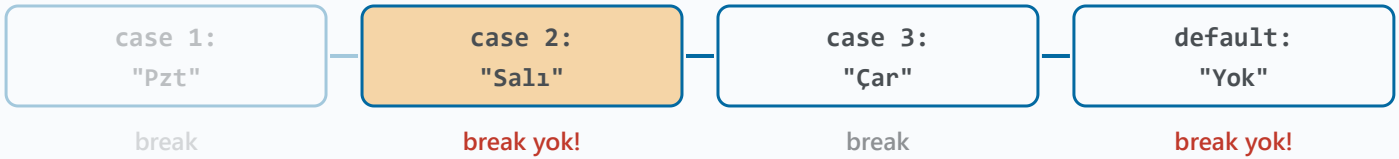
DEĞİŞKENLER

gun = **2**

ÇIKTI

SWITCH · KODU ADIM ADIM ÇALIŞTIR

Klasik switch: `break` Unutulursa



switch (gun) → 2: doğrudan `case 2` etiketine atlanır; case 1 hiç denetlenmez.

Adım 2 / 5

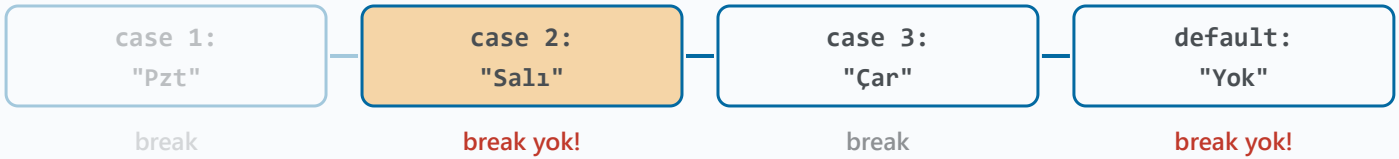
```
1 int gun = 2;
▶ switch (gun) {
3   case 1: System.out.println("Pzt"); break;
4   case 2: System.out.println("Salı");
5   case 3: System.out.println("Çar"); break;
6   default: System.out.println("Yok");
7 }
```

DEĞİŞKENLER

gun = **2**

ÇIKTI

Klasik switch: `break` Unutulursa



"Salı" yazdırılır. Bu case'te **break yok**!

Adım 3 / 5

```
1 int gun = 2;
2 switch (gun) {
3   case 1: System.out.println("Pzt"); break;
▶   case 2: System.out.println("Salı");
5   case 3: System.out.println("Çar"); break;
6   default: System.out.println("Yok");
7 }
```

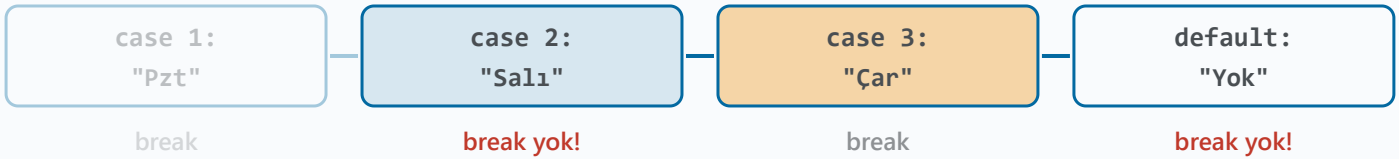
DEĞİŞKENLER

`gun = 2`

ÇIKTI

```
Salı
```

Klasik switch: `break` Unutulursa



Aşağı düşme (fall-through): etiket denetlenmeden sonraki case'in kodu da çalışır → "Çar".

Adım 4 / 5

```
1 int gun = 2;
2 switch (gun) {
3   case 1: System.out.println("Pzt"); break;
4   case 2: System.out.println("Salı");
▶   case 3: System.out.println("Çar"); break;
6   default: System.out.println("Yok");
7 }
```

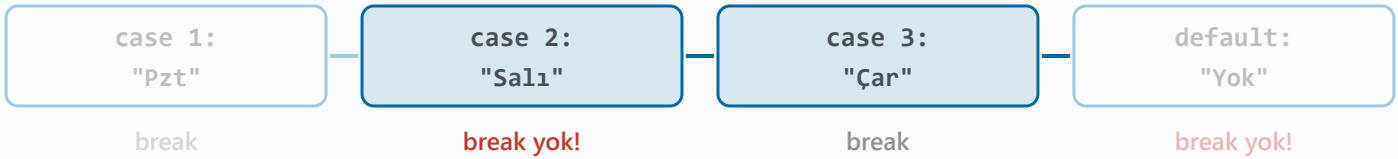
DEĞİŞKENLER

`gun = 2`

ÇIKTI

```
Salı
Çar
```

Klasik switch: `break` Unutulursa



break switch'ten çıkarır; default çalışmaz. Ok (`->`) biçimi bu hatayı önler.

Adım 5 / 5

```
1 int gun = 2;
2 switch (gun) {
3   case 1: System.out.println("Pzt"); break;
4   case 2: System.out.println("Salı");
▶   case 3: System.out.println("Çar"); break;
6   default: System.out.println("Yok");
7 }
```

DEĞİŞKENLER

`gun = 2`

ÇIKTI

```
Salı
Çar
```

SWITCH

if-else ve switch

Ölçüt	if-else	switch
Aralık koşulu	Uygun	Doğrudan uygun değil
Tek değere göre çok seçenek	Kullanılabilir	Genellikle daha açık
Karmaşık mantık	Uygun	Sınırlı
Her dal sonuç üretir	Atama gerekir	switch ifadesi değer döndürebilir

if-else: aralık koşulu



`not < 50` · `not < 85` · `else`

her dal bir sayı aralığını kapsar

switch: ayrık değerler



`case 1, 2, 3, 4, 5` — `"Hafta içi"`

`case 6, 7` — `"Hafta sonu"`

tek bir değişkenin sabit değerleri

ÖRNEK

Tek mi Çift mi?

```java
int sayi = 17;
if (sayi % 2 == 0)
    System.out.println("Çift");
else
    System.out.println("Tek");
```



- 2 ile bölümden kalan 0 ise sayı çifttir.
- Negatiflerde de doğru çalışır. Ama Java'da -7 % 2 = -1 olduğundan tek sayıyı `sayi % 2 == 1` ile sınamak yanlıştır.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)



İfade

```
sayi % 2 == 0
```

1. çalıştırma: sayi = **17**.

Adım 1 / 12

```
▶ int sayi = klavye.nextInt();
2 if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
5     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = **17**

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

17

int

İfade

17 % 2 == 0

1 == 0

Önce kalan: 17 % 2 = **1**.

Adım 2 / 12

```
1 int sayi = klavye.nextInt();
▶ if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
5     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = 17

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

17

int

İfade

1 == 0
→ false

1 == 0 → **false**.

Adım 3 / 12

```
1 int sayi = klavye.nextInt();
▶ if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
5     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = 17

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

17

int

İfade

→ "Tek"

Tek yazdırılır.

Adım 4 / 12

```
1 int sayi = klavye.nextInt();
2 if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
▶     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = 17

ÇIKTI

```
Tek
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

-4

int

İfade

sayi % 2 == 0

2. çalıştırma: sayi = **-4**.

Adım 5 / 12

```
▶ int sayi = klavye.nextInt();
2 if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
5     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = −4

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

-4

int

İfade

-4 % 2 == 0

0 == 0

Önce kalan: -4 % 2 = **0**.

Adım 6 / 12

```
1 int sayi = klavye.nextInt();
▶ if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
5     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = −4

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

-4

int

İfade

0 == 0
→ true

0 == 0 → **true**.

Adım 7 / 12

```
1 int sayi = klavye.nextInt();
▶ if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
5     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = −4

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

-4

int

İfade

→ "Çift"

Çift yazdırılır.

Adım 8 / 12

```
1 int sayi = klavye.nextInt();
2 if (sayi % 2 == 0)
▶     System.out.println("Çift");
4 else
5     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = −4

ÇIKTI

```
Çift
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

-7

int

İfade

sayi % 2 == 0

3. çalıştırma: sayi = **-7**.

Adım 9 / 12

```
▶ int sayi = klavye.nextInt();
2 if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
5     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = −7

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

-7

int

İfade

```
-7 % 2 == 0
  -1 == 0
```

Önce kalan: -7 % 2 = **-1** (Java'da kalanın işareti bölünenin işaretidir).

Adım 10 / 12

```
1 int sayi = klavye.nextInt();
▶ if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
5     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = −7

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

-7

int

İfade

-1 == 0
→ false

-1 == 0 → **false**.

Adım 11 / 12

```
1 int sayi = klavye.nextInt();
▶ if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
5     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = −7

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Tek mi Çift mi? sayi = 17, −4, −7

Bellek (değişkenler)

sayi

-7

int

İfade

→ "Tek"

Tek yazdırılır. -7 % 2 == 1 koşulu kullanılsaydı sonuç false olurdu!

Adım 12 / 12

```
1 int sayi = klavye.nextInt();
2 if (sayi % 2 == 0)
3     System.out.println("Çift");
4 else
▶     System.out.println("Tek");
```

DEĞİŞKENLER

sayi = −7

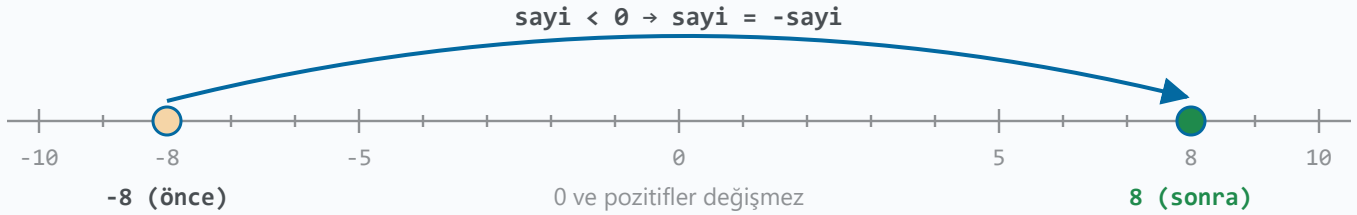
ÇIKTI

```
Tek
```

ÖRNEK

Mutlak Değer

```
int sayi = -8;
if (sayi < 0) {
    sayi = -sayi;
}
System.out.println(sayi);
```



- Sıfır ve pozitif değer değişmeden kalır.
- Integer.MIN_VALUE özel bir taşma durumudur.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE



1. çalıştırma: sayi = **-8**.

Adım 1 / 11

```
▶ int sayi = klavye.nextInt();
2 if (sayi < 0) {
3     sayi = -sayi;
4 }
5 System.out.println(sayi);
```

DEĞİŞKENLER

sayi = **-8**

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE

Bellek (değişkenler)



İfade

```
-8 < 0
→ true
```

Koşul: sayi < 0 → -8 < 0 → **true**.

Adım 2 / 11

```
1 int sayi = klavye.nextInt();
▶ if (sayi < 0) {
3     sayi = -sayi;
4 }
5 System.out.println(sayi);
```

DEĞİŞKENLER

sayi = **-8**

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE

Bellek (değişkenler)



sayi = -sayi → -(-8) = **8**.

Adım 3 / 11

```
1 int sayi = klavye.nextInt();
2 if (sayi < 0) {
▶     sayi = -sayi;
4 }
5 System.out.println(sayi);
```

DEĞİŞKENLER

sayi = 8

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE

Bellek (değişkenler)



8 yazdırılır.

Adım 4 / 11

```
1 int sayi = klavye.nextInt();
2 if (sayi < 0) {
3     sayi = -sayi;
4 }
▶ System.out.println(sayi);
```

DEĞİŞKENLER

sayi = **8**

ÇIKTI

```
8
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE

Bellek (değişkenler)



2. çalıştırma: sayi = **5**.

Adım 5 / 11

```
▶ int sayi = klavye.nextInt();
2 if (sayi < 0) {
3     sayi = -sayi;
4 }
5 System.out.println(sayi);
```

DEĞİŞKENLER

`sayi = 5`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE

Bellek (değişkenler)



İfade

```
5 < 0
→ false
```

Koşul: sayi < 0 → 5 < 0 → **false**. Blok atlanır, değer değişmez.

Adım 6 / 11

```
1 int sayi = klavye.nextInt();
▶ if (sayi < 0) {
3     sayi = -sayi;
4 }
5 System.out.println(sayi);
```

DEĞİŞKENLER

sayi = 5

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE



5 yazdırılır.

Adım 7 / 11

```
1 int sayi = klavye.nextInt();
2 if (sayi < 0) {
3     sayi = -sayi;
4 }
▶ System.out.println(sayi);
```

DEĞİŞKENLER

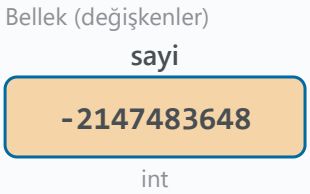
sayi = **5**

ÇIKTI

```
5
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE



3. çalıştırma: sayi = **-2147483648** (Integer.MIN_VALUE).

Adım 8 / 11

```
▶ int sayi = klavye.nextInt();
2 if (sayi < 0) {
3     sayi = -sayi;
4 }
5 System.out.println(sayi);
```

DEĞİŞKENLER

sayi = −2147483648

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE

Bellek (değişkenler)

sayi

-2147483648

int

İfade

-2147483648 < 0

→ true

Koşul: sayi < 0 → -2147483648 < 0 → **true**.

Adım 9 / 11

```
1 int sayi = klavye.nextInt();
▶ if (sayi < 0) {
3     sayi = -sayi;
4 }
5 System.out.println(sayi);
```

DEĞİŞKENLER

sayi = −2147483648

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE

Bellek (değişkenler)

sayi

-2147483648

int

-(-2147483648) = 2147483648 int aralığına sığmaz → **taşma**: sonuç yine -2147483648!

Adım 10 / 11

```
1 int sayi = klavye.nextInt();
2 if (sayi < 0) {
▶     sayi = -sayi;
4 }
5 System.out.println(sayi);
```

DEĞİŞKENLER

`sayi = -2147483648`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Mutlak Değer: sayi = −8, 5, MIN_VALUE

Bellek (değişkenler)

sayi

-2147483648

int

-2147483648 yazdırılır. Mutlak değer negatif çıktı: özel durum.

Adım 11 / 11

```
1 int sayi = klavye.nextInt();
2 if (sayi < 0) {
3     sayi = -sayi;
4 }
▶ System.out.println(sayi);
```

DEĞİŞKENLER

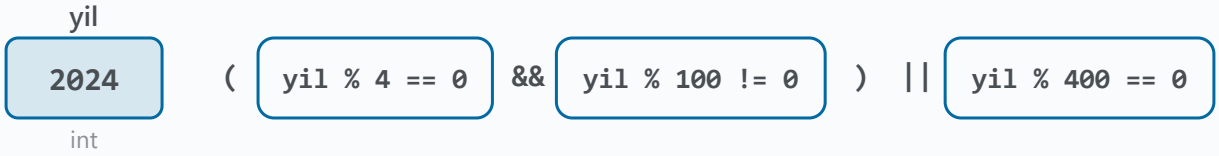
sayi = **-2147483648**

ÇIKTI

```
-2147483648
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



1. çalıştırma: yil = 2024.

Adım 1 / 16

```
▶ int yil = klavye.nextInt();
2 boolean artik =
3     (yil % 4 == 0 && yil % 100 != 0)
4     || yil % 400 == 0;
5 if (artik)
6     System.out.println(yil + " artık yıl");
7 else
8     System.out.println(yil + " normal yıl");
```

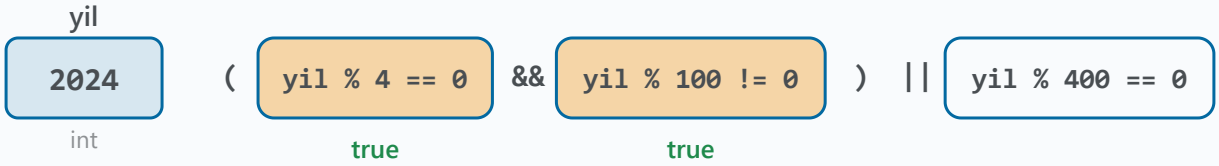
DEĞİŞKENLER

yil = 2024

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



2024 % 4 = 0 → true; 2024 % 100 = 24 → yil % 100 != 0 → true. Parantez içi: **true**.

Adım 2 / 16

```
1  int yil = klavye.nextInt();
▶  boolean artik =
▶      (yil % 4 == 0 && yil % 100 != 0)
4      || yil % 400 == 0;
5  if (artik)
6      System.out.println(yil + " artık yıl");
7  else
8      System.out.println(yil + " normal yıl");
```

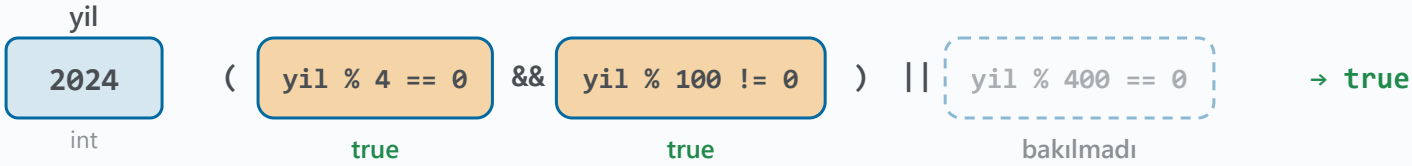
DEĞİŞKENLER

`yil = 2024`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



Sol taraf true → `||` kısa devre: 400 kuralına bakılmaz. artik = **true**.

Adım 3 / 16

```
1  int yil = klavye.nextInt();
▶  boolean artik =
3      (yil % 4 == 0 && yil % 100 != 0)
▶      || yil % 400 == 0;
5  if (artik)
6      System.out.println(yil + " artık yıl");
7  else
8      System.out.println(yil + " normal yıl");
```

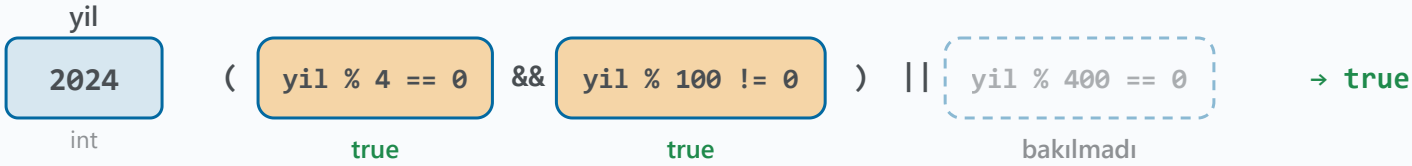
DEĞİŞKENLER

`yil = 2024` `artik = true`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



2024 artık yıl.

Adım 4 / 16

```
1  int yil = klavye.nextInt();
2  boolean artik =
3      (yil % 4 == 0 && yil % 100 != 0)
4      || yil % 400 == 0;
5  if (artik)
▶      System.out.println(yil + " artık yıl");
7  else
8      System.out.println(yil + " normal yıl");
```

DEĞİŞKENLER

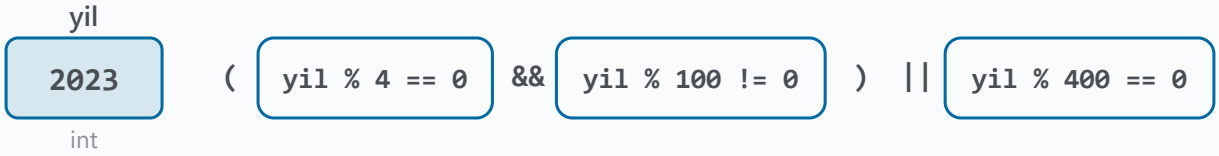
`yil = 2024` `artik = true`

ÇIKTI

```
2024 artık yıl
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



2. çalıştırma: yil = 2023.

Adım 5 / 16

```
▶ int yil = klavye.nextInt();
2 boolean artik =
3     (yil % 4 == 0 && yil % 100 != 0)
4     || yil % 400 == 0;
5 if (artik)
6     System.out.println(yil + " artık yıl");
7 else
8     System.out.println(yil + " normal yıl");
```

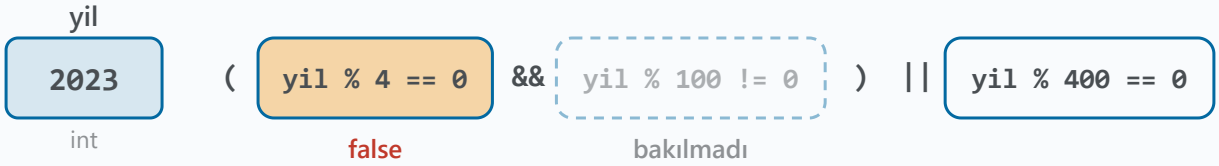
DEĞİŞKENLER

yil = 2023

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



2023 % 4 = 3 → false; && kısa devre yapar. Parantez içi: **false**.

Adım 6 / 16

```
1  int yil = klavye.nextInt();
▶  boolean artik =
▶      (yil % 4 == 0 && yil % 100 != 0)
4      || yil % 400 == 0;
5  if (artik)
6      System.out.println(yil + " artık yıl");
7  else
8      System.out.println(yil + " normal yıl");
```

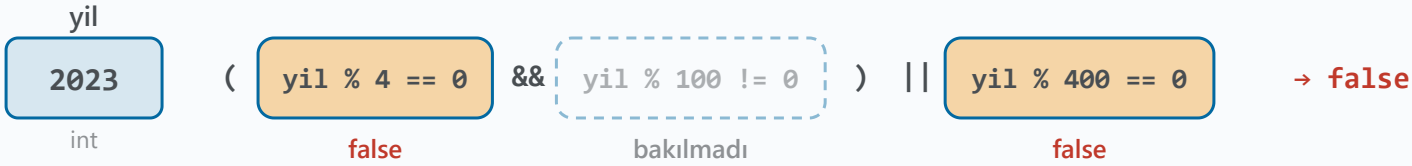
DEĞİŞKENLER

`yil = 2023`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



Sol taraf false → sağa bakılır: 2023 % 400 = 23 → **false**. artik = **false**.

Adım 7 / 16

```
1  int yil = klavye.nextInt();
▶  boolean artik =
3      (yil % 4 == 0 && yil % 100 != 0)
▶      || yil % 400 == 0;
5  if (artik)
6      System.out.println(yil + " artık yıl");
7  else
8      System.out.println(yil + " normal yıl");
```

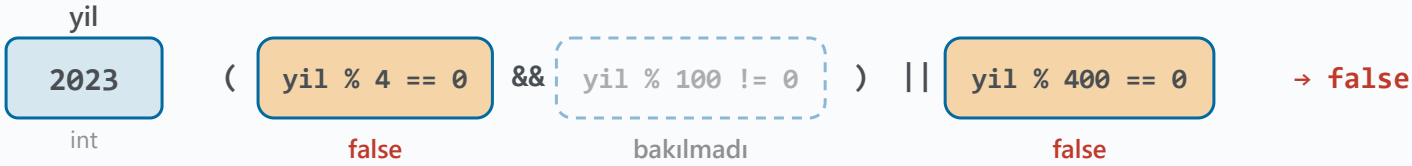
DEĞİŞKENLER

`yil = 2023` `artik = false`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



2023 normal yıl.

Adım 8 / 16

```
1 int yil = klavye.nextInt();
2 boolean artik =
3     (yil % 4 == 0 && yil % 100 != 0)
4     || yil % 400 == 0;
5 if (artik)
6     System.out.println(yil + " artık yıl");
7 else
▶     System.out.println(yil + " normal yıl");
```

DEĞİŞKENLER

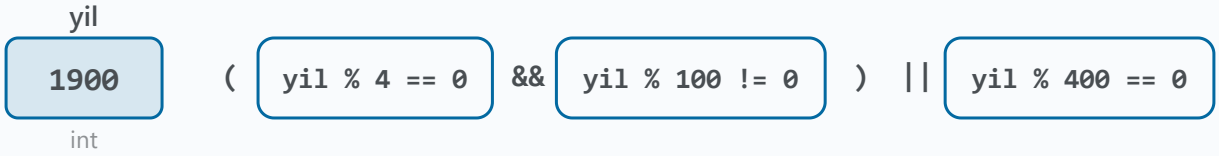
`yil = 2023` `artik = false`

ÇIKTI

```
2023 normal yıl
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



3. çalıştırma: yil = 1900.

Adım 9 / 16

```
▶ int yil = klavye.nextInt();
2 boolean artik =
3     (yil % 4 == 0 && yil % 100 != 0)
4     || yil % 400 == 0;
5 if (artik)
6     System.out.println(yil + " artık yıl");
7 else
8     System.out.println(yil + " normal yıl");
```

DEĞİŞKENLER

yil = 1900

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



1900 % 4 = 0 → true; 1900 % 100 = 0 → yil % 100 != 0 → false. Parantez içi: **false**.

Adım 10 / 16

```
1 int yil = klavye.nextInt();
▶ boolean artik =
▶     (yil % 4 == 0 && yil % 100 != 0)
4     || yil % 400 == 0;
5 if (artik)
6     System.out.println(yil + " artık yıl");
7 else
8     System.out.println(yil + " normal yıl");
```

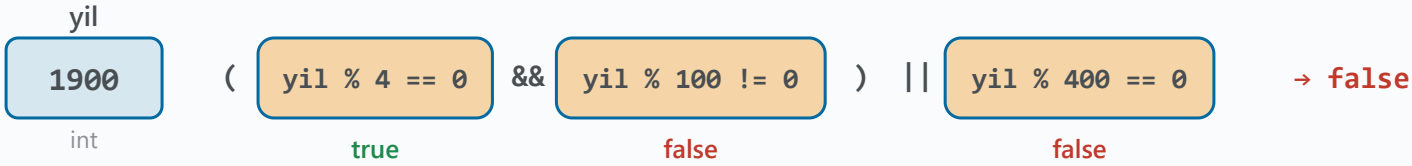
DEĞİŞKENLER

`yil = 1900`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



Sol taraf false → sağa bakılır: 1900 % 400 = 300 → **false**. artik = **false**.

Adım 11 / 16

```
1  int yil = klavye.nextInt();
▶  boolean artik =
3      (yil % 4 == 0 && yil % 100 != 0)
▶      || yil % 400 == 0;
5  if (artik)
6      System.out.println(yil + " artık yıl");
7  else
8      System.out.println(yil + " normal yıl");
```

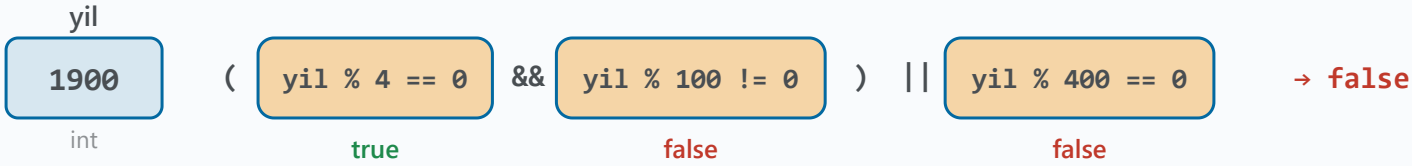
DEĞİŞKENLER

`yil = 1900` `artik = false`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



1900 normal yıl. 100'e bölünen yüzyıllar, 400'e de bölünmedikçe artık değildir.

Adım 12 / 16

```
1 int yil = klavye.nextInt();
2 boolean artik =
3     (yil % 4 == 0 && yil % 100 != 0)
4     || yil % 400 == 0;
5 if (artik)
6     System.out.println(yil + " artık yıl");
7 else
▶     System.out.println(yil + " normal yıl");
```

DEĞİŞKENLER

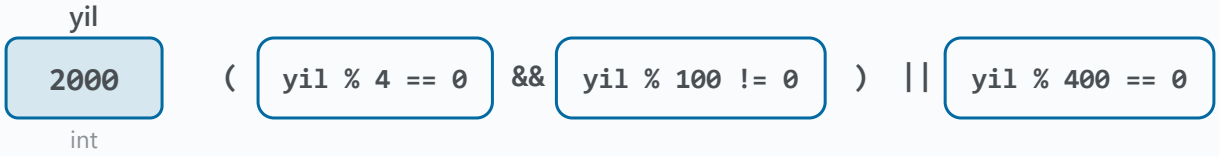
`yil = 1900` `artik = false`

ÇIKTI

```
1900 normal yıl
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



4. çalıştırma: yil = 2000.

Adım 13 / 16

```
▶ int yil = klavye.nextInt();
2 boolean artik =
3     (yil % 4 == 0 && yil % 100 != 0)
4     || yil % 400 == 0;
5 if (artik)
6     System.out.println(yil + " artık yıl");
7 else
8     System.out.println(yil + " normal yıl");
```

DEĞİŞKENLER

yil = 2000

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



2000 % 4 = 0 → true; 2000 % 100 = 0 → yil % 100 != 0 → false. Parantez içi: **false**.

Adım 14 / 16

```
1  int yil = klavye.nextInt();
▶  boolean artik =
▶      (yil % 4 == 0 && yil % 100 != 0)
4      || yil % 400 == 0;
5  if (artik)
6      System.out.println(yil + " artık yıl");
7  else
8      System.out.println(yil + " normal yıl");
```

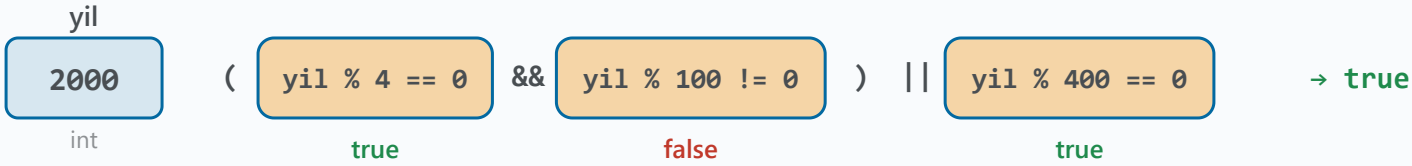
DEĞİŞKENLER

`yil = 2000`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



Sol taraf false → sağa bakılır: 2000 % 400 = 0 → **true**. artik = **true**.

Adım 15 / 16

```
1  int yil = klavye.nextInt();
▶  boolean artik =
3      (yil % 4 == 0 && yil % 100 != 0)
▶      || yil % 400 == 0;
5  if (artik)
6      System.out.println(yil + " artık yıl");
7  else
8      System.out.println(yil + " normal yıl");
```

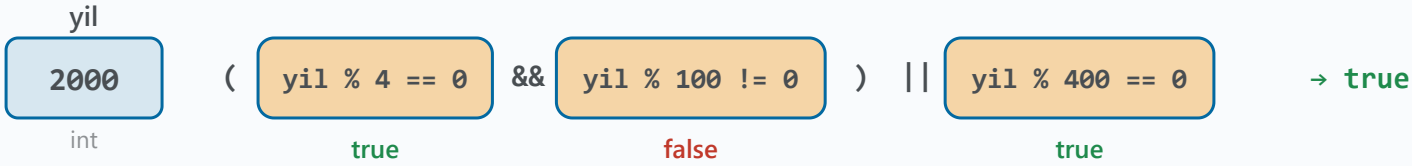
DEĞİŞKENLER

`yil = 2000` `artik = true`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Artık Yıl: 2024, 2023, 1900, 2000



2000 artık yıl.

Adım 16 / 16

```
1 int yil = klavye.nextInt();
2 boolean artik =
3     (yil % 4 == 0 && yil % 100 != 0)
4     || yil % 400 == 0;
5 if (artik)
▶     System.out.println(yil + " artık yıl");
7 else
8     System.out.println(yil + " normal yıl");
```

DEĞİŞKENLER

`yil = 2000` `artik = true`

ÇIKTI

```
2000 artık yıl
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



1.0x² + (-3.0)x + 2.0

1. çalıştırma: a = 1.0, b = -3.0, c = 2.0.

Adım 1 / 13

```
▶ double a = klavye.nextDouble();
▶ double b = klavye.nextDouble();
▶ double c = klavye.nextDouble();
4 double d = b * b - 4 * a * c;
5 if (d > 0) {
6   double x1 = (-b + Math.sqrt(d)) / (2 * a);
7   double x2 = (-b - Math.sqrt(d)) / (2 * a);
8   System.out.println(x1 + " ve " + x2);
9 } else if (d == 0) {
10  System.out.println(-b / (2 * a));
11 } else {
12  System.out.println("Reel kök yok");
13 }
```

DEĞİŞKENLER

a = 1.0 | b = -3.0 | c = 2.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



$1.0x^2 + (-3.0)x + 2.0$

Δ = 1.0

d = Δ = $b^2 - 4ac$ = 9.0 − 8.0 = **1.0**.

Adım 2 / 13

```
1  double a = klavye.nextDouble();
2  double b = klavye.nextDouble();
3  double c = klavye.nextDouble();
▶ double d = b * b - 4 * a * c;
5  if (d > 0) {
6    double x1 = (-b + Math.sqrt(d)) / (2 * a);
7    double x2 = (-b - Math.sqrt(d)) / (2 * a);
8    System.out.println(x1 + " ve " + x2);
9  } else if (d == 0) {
10   System.out.println(-b / (2 * a));
11 } else {
12   System.out.println("Reel kök yok");
13 }
```

DEĞİŞKENLER

a = 1.0 | b = -3.0 | c = 2.0 | d = 1.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



1.0x² + (-3.0)x + 2.0

Δ > 0 → iki kök

1.0 > 0 → true: iki farklı reel kök.

Adım 3 / 13

```
1  double a = klavye.nextDouble();
2  double b = klavye.nextDouble();
3  double c = klavye.nextDouble();
4  double d = b * b - 4 * a * c;
▶  if (d > 0) {
6    double x1 = (-b + Math.sqrt(d)) / (2 * a);
7    double x2 = (-b - Math.sqrt(d)) / (2 * a);
8    System.out.println(x1 + " ve " + x2);
9  } else if (d == 0) {
10   System.out.println(-b / (2 * a));
11 } else {
12   System.out.println("Reel kök yok");
13 }
```

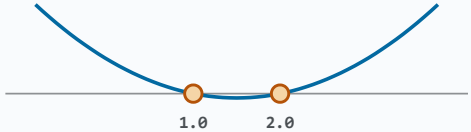
DEĞİŞKENLER

a = 1.0 | b = -3.0 | c = 2.0 | d = 1.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



1.0x² + (-3.0)x + 2.0

iki kök

√Δ = 1.0; x1 = (3.0 + 1.0) / 2.0 = 2.0, x2 = 1.0.

Adım 4 / 13

```
1 double a = klavye.nextDouble();
2 double b = klavye.nextDouble();
3 double c = klavye.nextDouble();
4 double d = b * b - 4 * a * c;
5 if (d > 0) {
▶   double x1 = (-b + Math.sqrt(d)) / (2 * a);
▶   double x2 = (-b - Math.sqrt(d)) / (2 * a);
8   System.out.println(x1 + " ve " + x2);
9 } else if (d == 0) {
10  System.out.println(-b / (2 * a));
11 } else {
12  System.out.println("Reel kök yok");
13 }
```

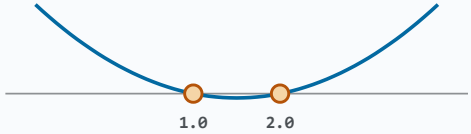
DEĞİŞKENLER

a = 1.0 | b = -3.0 | c = 2.0 | d = 1.0 | x1 = 2.0 | x2 = 1.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



$1.0x^2 + (-3.0)x + 2.0$

iki kök

2.0 ve 1.0: parabol x eksenini iki noktada keser.

Adım 5 / 13

```
1  double a = klavye.nextDouble();
2  double b = klavye.nextDouble();
3  double c = klavye.nextDouble();
4  double d = b * b - 4 * a * c;
5  if (d > 0) {
6    double x1 = (-b + Math.sqrt(d)) / (2 * a);
7    double x2 = (-b - Math.sqrt(d)) / (2 * a);
▶    System.out.println(x1 + " ve " + x2);
9  } else if (d == 0) {
10   System.out.println(-b / (2 * a));
11 } else {
12   System.out.println("Reel kök yok");
13 }
```

DEĞİŞKENLER

a = 1.0 | b = -3.0 | c = 2.0 | d = 1.0 | x1 = 2.0 | x2 = 1.0

ÇIKTI

```
2.0 ve 1.0
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



$1.0x^2 + (-2.0)x + 1.0$

2. çalıştırma: a = 1.0, b = -2.0, c = 1.0.

Adım 6 / 13

```
▶ double a = klavye.nextDouble();
▶ double b = klavye.nextDouble();
▶ double c = klavye.nextDouble();
4 double d = b * b - 4 * a * c;
5 if (d > 0) {
6   double x1 = (-b + Math.sqrt(d)) / (2 * a);
7   double x2 = (-b - Math.sqrt(d)) / (2 * a);
8   System.out.println(x1 + " ve " + x2);
9 } else if (d == 0) {
10  System.out.println(-b / (2 * a));
11 } else {
12  System.out.println("Reel kök yok");
13 }
```

DEĞİŞKENLER

a = 1.0 b = -2.0 c = 1.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



$1.0x^2 + (-2.0)x + 1.0$

Δ = 0.0

d = Δ = $b^2 - 4ac = 4.0 - 4.0 = \mathbf{0.0}$.

Adım 7 / 13

```
1  double a = klavye.nextDouble();
2  double b = klavye.nextDouble();
3  double c = klavye.nextDouble();
▶ double d = b * b - 4 * a * c;
5  if (d > 0) {
6    double x1 = (-b + Math.sqrt(d)) / (2 * a);
7    double x2 = (-b - Math.sqrt(d)) / (2 * a);
8    System.out.println(x1 + " ve " + x2);
9  } else if (d == 0) {
10   System.out.println(-b / (2 * a));
11 } else {
12   System.out.println("Reel kök yok");
13 }
```

DEĞİŞKENLER

a = 1.0 | b = -2.0 | c = 1.0 | d = 0.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



1.0x² + (-2.0)x + 1.0

Δ = 0 → tek kök

Δ > 0 false; Δ == 0 → **true**: tek (çakışık) kök.

Adım 8 / 13

```
1  double a = klavye.nextDouble();
2  double b = klavye.nextDouble();
3  double c = klavye.nextDouble();
4  double d = b * b - 4 * a * c;
▶  if (d > 0) {
6    double x1 = (-b + Math.sqrt(d)) / (2 * a);
7    double x2 = (-b - Math.sqrt(d)) / (2 * a);
8    System.out.println(x1 + " ve " + x2);
▶  } else if (d == 0) {
10   System.out.println(-b / (2 * a));
11 } else {
12   System.out.println("Reel kök yok");
13 }
```

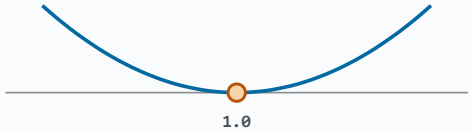
DEĞİŞKENLER

a = 1.0 | b = -2.0 | c = 1.0 | d = 0.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



1.0x² + (-2.0)x + 1.0

tek kök

x = −b / 2a = 2.0 / 2.0 = **1.0**: parabol eksene teğet.

Adım 9 / 13

```
1  double a = klavye.nextDouble();
2  double b = klavye.nextDouble();
3  double c = klavye.nextDouble();
4  double d = b * b - 4 * a * c;
5  if (d > 0) {
6    double x1 = (-b + Math.sqrt(d)) / (2 * a);
7    double x2 = (-b - Math.sqrt(d)) / (2 * a);
8    System.out.println(x1 + " ve " + x2);
9  } else if (d == 0) {
▶    System.out.println(-b / (2 * a));
11 } else {
12   System.out.println("Reel kök yok");
13 }
```

DEĞİŞKENLER

a = 1.0 | b = -2.0 | c = 1.0 | d = 0.0

ÇIKTI

```
1.0
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



$1.0x^2 + (0.0)x + 1.0$

3. çalıştırma: a = 1.0, b = 0.0, c = 1.0.

Adım 10 / 13

```
▶ double a = klavye.nextDouble();
▶ double b = klavye.nextDouble();
▶ double c = klavye.nextDouble();
4 double d = b * b - 4 * a * c;
5 if (d > 0) {
6   double x1 = (-b + Math.sqrt(d)) / (2 * a);
7   double x2 = (-b - Math.sqrt(d)) / (2 * a);
8   System.out.println(x1 + " ve " + x2);
9 } else if (d == 0) {
10  System.out.println(-b / (2 * a));
11 } else {
12  System.out.println("Reel kök yok");
13 }
```

DEĞİŞKENLER

a = 1.0 | b = 0.0 | c = 1.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



$1.0x^2 + (0.0)x + 1.0$

Δ = -4.0

d = Δ = b^2 − 4ac = 0.0 − 4.0 = **-4.0**.

Adım 11 / 13

```
1  double a = klavye.nextDouble();
2  double b = klavye.nextDouble();
3  double c = klavye.nextDouble();
▶ 4  double d = b * b - 4 * a * c;
5  if (d > 0) {
6    double x1 = (-b + Math.sqrt(d)) / (2 * a);
7    double x2 = (-b - Math.sqrt(d)) / (2 * a);
8    System.out.println(x1 + " ve " + x2);
9  } else if (d == 0) {
10   System.out.println(-b / (2 * a));
11 } else {
12   System.out.println("Reel kök yok");
13 }
```

DEĞİŞKENLER

a = 1.0 | b = 0.0 | c = 1.0 | d = -4.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



$1.0x^2 + (0.0)x + 1.0$

$\Delta < 0$

-4.0 > 0 false, -4.0 == 0 false → else.

Adım 12 / 13

```
1  double a = klavye.nextDouble();
2  double b = klavye.nextDouble();
3  double c = klavye.nextDouble();
4  double d = b * b - 4 * a * c;
▶  if (d > 0) {
6    double x1 = (-b + Math.sqrt(d)) / (2 * a);
7    double x2 = (-b - Math.sqrt(d)) / (2 * a);
8    System.out.println(x1 + " ve " + x2);
▶  } else if (d == 0) {
10   System.out.println(-b / (2 * a));
11 } else {
12   System.out.println("Reel kök yok");
13 }
```

DEĞİŞKENLER

a = 1.0 | b = 0.0 | c = 1.0 | d = -4.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

İkinci Dereceden Denklem: Δ > 0, Δ = 0, Δ < 0



$1.0x^2 + (0.0)x + 1.0$

kök yok

Reel kök yok: parabol ekseni kesmez. Math.sqrt(negatif) NaN verirdi; else bunu önler.

Adım 13 / 13

```
1  double a = klavye.nextDouble();
2  double b = klavye.nextDouble();
3  double c = klavye.nextDouble();
4  double d = b * b - 4 * a * c;
5  if (d > 0) {
6    double x1 = (-b + Math.sqrt(d)) / (2 * a);
7    double x2 = (-b - Math.sqrt(d)) / (2 * a);
8    System.out.println(x1 + " ve " + x2);
9  } else if (d == 0) {
10   System.out.println(-b / (2 * a));
11 } else {
   ▶ System.out.println("Reel kök yok");
13 }
```

DEĞİŞKENLER

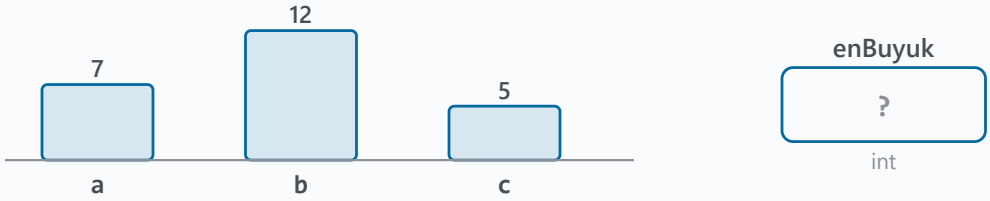
a = 1.0 | b = 0.0 | c = 1.0 | d = -4.0

ÇIKTI

```
Reel kök yok
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi



1. çalıştırma: 7, 12, 5.

Adım 1 / 17

```
▶ int a = klavye.nextInt();
▶ int b = klavye.nextInt();
▶ int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

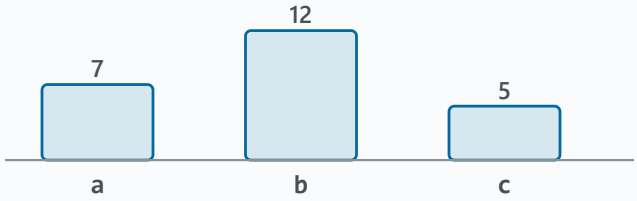
a = 7 | b = 12 | c = 5

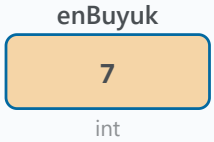
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





Aday: enBuyuk = a = 7.

Adım 2 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
▶ int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

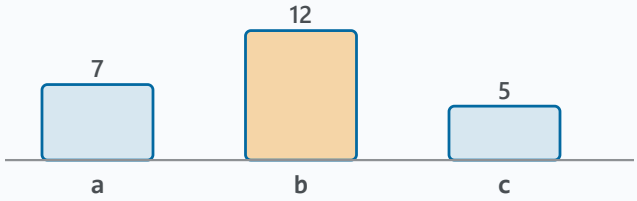
a = 7 | b = 12 | c = 5 | enBuyuk = 7

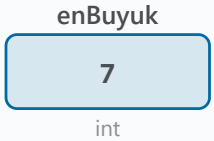
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





b > enBuyuk → 12 > 7 → **true**.

Adım 3 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
▶ if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

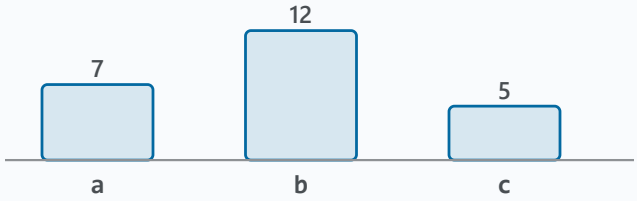
`a = 7` `b = 12` `c = 5` `enBuyuk = 7`

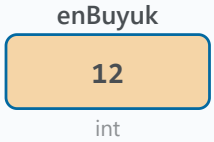
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





Yeni aday: enBuyuk = b = 12.

Adım 4 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
▶     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

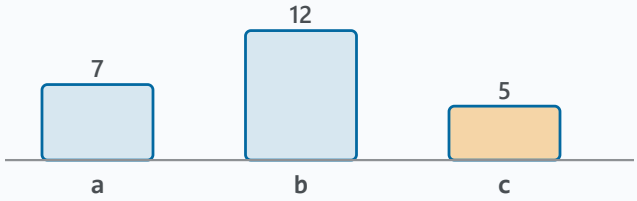
a = 7 | b = 12 | c = 5 | enBuyuk = 12

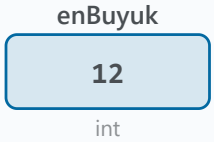
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





c > enBuyuk → 5 > 12 → **false**; aday değişmez.

Adım 5 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
▶ if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

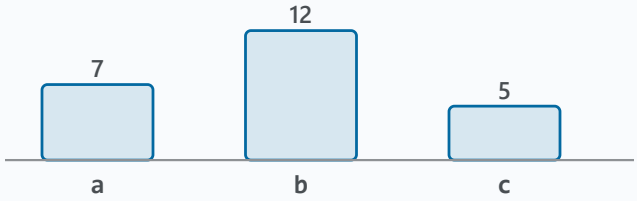
a = 7 | b = 12 | c = 5 | enBuyuk = 12

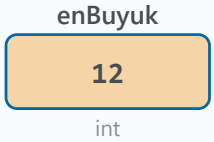
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





En büyük: 12.

Adım 6 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
▶ System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

`a = 7` `b = 12` `c = 5` `enBuyuk = 12`

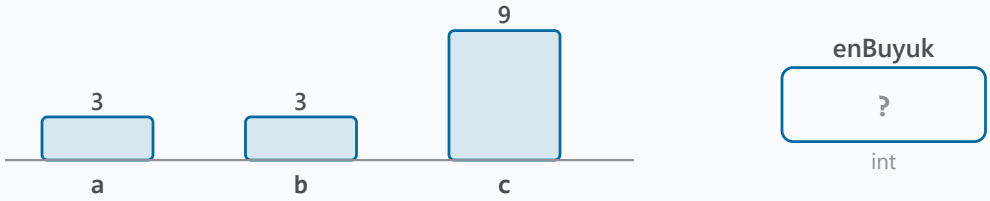
ÇIKTI

```
En büyük: 12
```

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi



2. çalıştırma: 3, 3, 9.

Adım 7 / 17

```
▶ int a = klavye.nextInt();
▶ int b = klavye.nextInt();
▶ int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

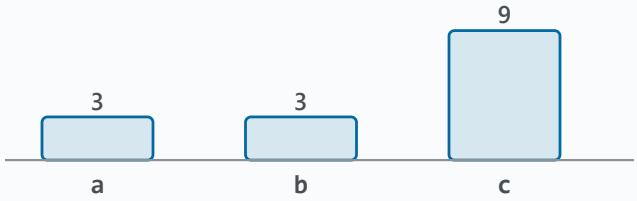
a = 3 b = 3 c = 9

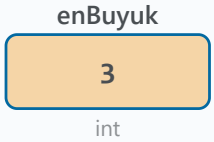
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





Aday: enBuyuk = a = 3.

Adım 8 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
▶ int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

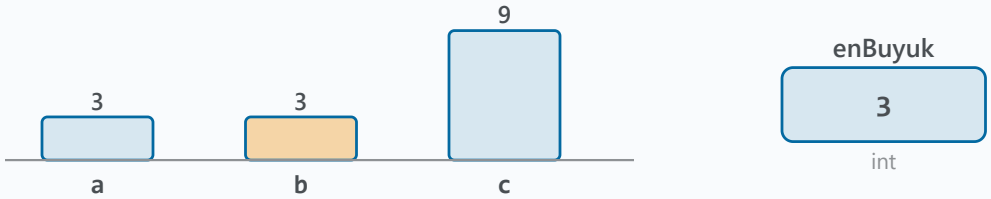
a = 3 | b = 3 | c = 9 | enBuyuk = 3

ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi



b > enBuyuk → 3 > 3 → **false**; aday değişmez. Eşit değer adayı değiştirmez.

Adım 9 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
▶ if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

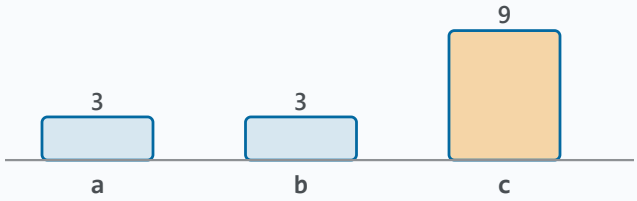
a = 3 | b = 3 | c = 9 | enBuyuk = 3

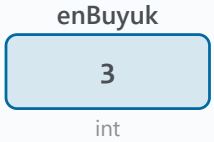
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





c > enBuyuk → 9 > 3 → **true**.

Adım 10 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
▶ if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

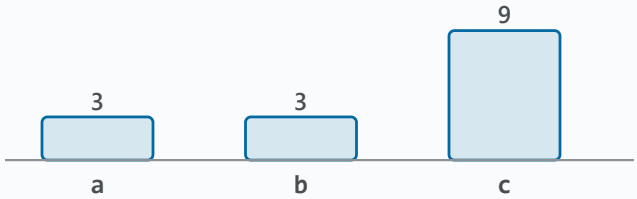
`a = 3` `b = 3` `c = 9` `enBuyuk = 3`

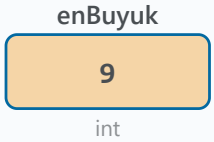
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





Yeni aday: enBuyuk = c = 9.

Adım 11 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
▶     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

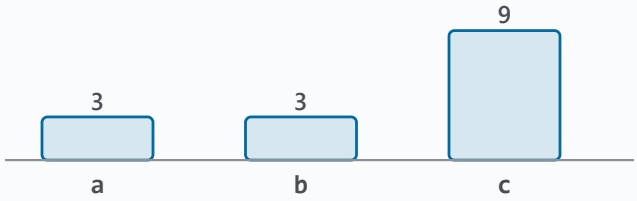
`a = 3` `b = 3` `c = 9` `enBuyuk = 9`

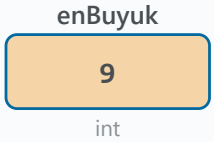
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





En büyük: 9.

Adım 12 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
▶ System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

`a = 3` `b = 3` `c = 9` `enBuyuk = 9`

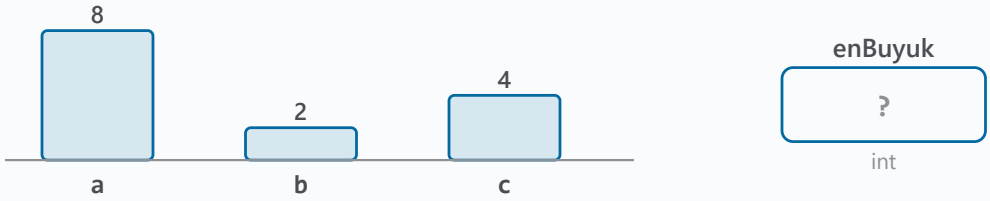
ÇIKTI

```
En büyük: 9
```

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi



3. çalıştırma: 8, 2, 4.

Adım 13 / 17

```
▶ int a = klavye.nextInt();
▶ int b = klavye.nextInt();
▶ int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

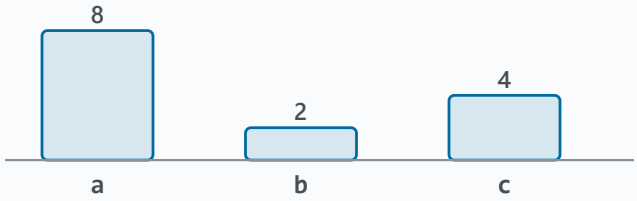
a = 8 b = 2 c = 4

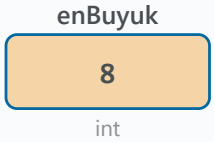
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





Aday: enBuyuk = a = 8.

Adım 14 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
▶ int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

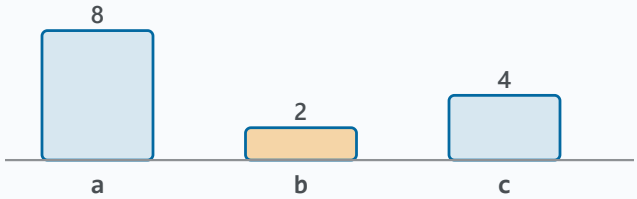
a = 8 | b = 2 | c = 4 | enBuyuk = 8

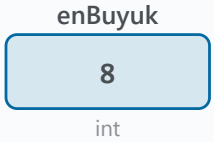
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





b > enBuyuk → 2 > 8 → **false**; aday değişmez.

Adım 15 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
▶ if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

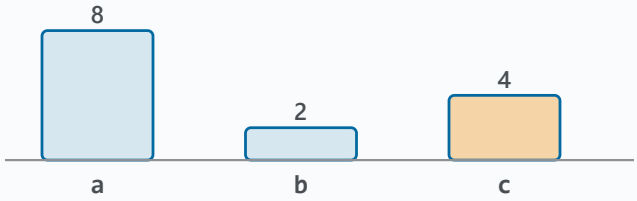
`a = 8` `b = 2` `c = 4` `enBuyuk = 8`

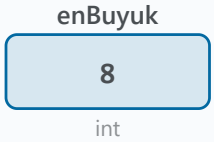
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





c > enBuyuk → 4 > 8 → **false**; aday değişmez.

Adım 16 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
▶ if (c > enBuyuk)
8     enBuyuk = c;
9 System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

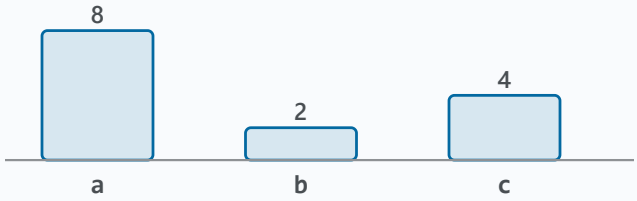
a = 8 | b = 2 | c = 4 | enBuyuk = 8

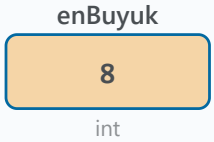
ÇIKTI

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

Üç Sayının En Büyüğü: 3 farklı girdi





En büyük: 8.

Adım 17 / 17

```
1 int a = klavye.nextInt();
2 int b = klavye.nextInt();
3 int c = klavye.nextInt();
4 int enBuyuk = a;
5 if (b > enBuyuk)
6     enBuyuk = b;
7 if (c > enBuyuk)
8     enBuyuk = c;
▶ System.out.println("En büyük: " + enBuyuk);
```

DEĞİŞKENLER

`a = 8` `b = 2` `c = 4` `enBuyuk = 8`

ÇIKTI

```
En büyük: 8
```

Kısa yol: `Math.max(a, Math.max(b, c))`. "Aday + karşılaştır" kalıbı döngüyle genellenebilir.

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

switch ile Hesap Makinesi: *, /0, %



1. çalıştırma: kullanıcı `8.0 * 2.5` girer; next() işareti dizgi olarak okur.

Adım 1 / 10

```
▶ double x = klavye.nextDouble();
▶ String op = klavye.next();
▶ double y = klavye.nextDouble();
4  switch (op) {
5    case "+" -> System.out.println(x + y);
6    case "-" -> System.out.println(x - y);
7    case "*" -> System.out.println(x * y);
8    case "/" -> {
9      if (y == 0) System.out.println("Hata!");
10     else System.out.println(x / y);
11   }
12   default -> System.out.println("Geçersiz");
13 }
```

DEĞİŞKENLER

x = 8.0 | op = "*" | y = 2.5

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

switch ile Hesap Makinesi: *, /0, %



switch (op) → "*": eşleşen **case "*"**.

Adım 2 / 10

```
1  double x = klavye.nextDouble();
2  String op = klavye.next();
3  double y = klavye.nextDouble();
▶ switch (op) {
5    case "+" -> System.out.println(x + y);
6    case "-" -> System.out.println(x - y);
7    case "*" -> System.out.println(x * y);
8    case "/" -> {
9      if (y == 0) System.out.println("Hata!");
10     else System.out.println(x / y);
11   }
12   default -> System.out.println("Geçersiz");
13 }
```

DEĞİŞKENLER

`x = 8.0` `op = "*"` `y = 2.5`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

switch ile Hesap Makinesi: *, /0, %



8.0 * 2.5 = **20.0** yazdırılır. Diğer case'ler çalışmaz.

Adım 3 / 10

```
1  double x = klavye.nextDouble();
2  String op = klavye.next();
3  double y = klavye.nextDouble();
4  switch (op) {
5    case "+" -> System.out.println(x + y);
6    case "-" -> System.out.println(x - y);
▶    case "*" -> System.out.println(x * y);
8    case "/" -> {
9      if (y == 0) System.out.println("Hata!");
10     else System.out.println(x / y);
11   }
12   default -> System.out.println("Geçersiz");
13 }
```

DEĞİŞKENLER

`x = 8.0` `op = "*"` `y = 2.5`

ÇIKTI

```
20.0
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

switch ile Hesap Makinesi: *, /0, %



2. çalıştırma: kullanıcı `7.0 / 0.0` girer; next() işareti dizgi olarak okur.

Adım 4 / 10

```
▶ double x = klavye.nextDouble();
▶ String op = klavye.next();
▶ double y = klavye.nextDouble();
4  switch (op) {
5    case "+" -> System.out.println(x + y);
6    case "-" -> System.out.println(x - y);
7    case "*" -> System.out.println(x * y);
8    case "/" -> {
9      if (y == 0) System.out.println("Hata!");
10     else System.out.println(x / y);
11   }
12   default -> System.out.println("Geçersiz");
13 }
```

DEĞİŞKENLER

x = 7.0 | op = "/" | y = 0.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

switch ile Hesap Makinesi: *, /0, %



switch (op) → "/": eşleşen **case "/"**.

Adım 5 / 10

```
1  double x = klavye.nextDouble();
2  String op = klavye.next();
3  double y = klavye.nextDouble();
▶  switch (op) {
5    case "+" -> System.out.println(x + y);
6    case "-" -> System.out.println(x - y);
7    case "*" -> System.out.println(x * y);
8    case "/" -> {
9      if (y == 0) System.out.println("Hata!");
10     else System.out.println(x / y);
11   }
12   default -> System.out.println("Geçersiz");
13 }
```

DEĞİŞKENLER

x = **7.0** | op = **"/"** | y = **0.0**

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

switch ile Hesap Makinesi: *, /0, %



Bölmeden önce denetim: y == 0 → **true**. (double'da 7.0 / 0 hata vermez, Infinity üretir; bu yüzden kontrol ederiz.)

Adım 6 / 10

```
1  double x = klavye.nextDouble();
2  String op = klavye.next();
3  double y = klavye.nextDouble();
4  switch (op) {
5    case "+" -> System.out.println(x + y);
6    case "-" -> System.out.println(x - y);
7    case "*" -> System.out.println(x * y);
8    case "/" -> {
▶      if (y == 0) System.out.println("Hata!");
10     else System.out.println(x / y);
11   }
12   default -> System.out.println("Geçersiz");
13 }
```

DEĞİŞKENLER

`x = 7.0` `op = "/"` `y = 0.0`

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

switch ile Hesap Makinesi: *, /0, %



Hata! yazdırılır.

Adım 7 / 10

```
1  double x = klavye.nextDouble();
2  String op = klavye.next();
3  double y = klavye.nextDouble();
4  switch (op) {
5    case "+" -> System.out.println(x + y);
6    case "-" -> System.out.println(x - y);
7    case "*" -> System.out.println(x * y);
8    case "/" -> {
▶      if (y == 0) System.out.println("Hata!");
10     else System.out.println(x / y);
11   }
12   default -> System.out.println("Geçersiz");
13 }
```

DEĞİŞKENLER

x = 7.0 | op = "/" | y = 0.0

ÇIKTI

```
Hata!
```

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

switch ile Hesap Makinesi: *, /0, %



3. çalıştırma: kullanıcı `5.0 % 3.0` girer; next() işareti dizgi olarak okur.

Adım 8 / 10

```
▶ double x = klavye.nextDouble();
▶ String op = klavye.next();
▶ double y = klavye.nextDouble();
4 switch (op) {
5   case "+" -> System.out.println(x + y);
6   case "-" -> System.out.println(x - y);
7   case "*" -> System.out.println(x * y);
8   case "/" -> {
9     if (y == 0) System.out.println("Hata!");
10    else System.out.println(x / y);
11  }
12  default -> System.out.println("Geçersiz");
13 }
```

DEĞİŞKENLER

x = 5.0 | op = "%" | y = 3.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

switch ile Hesap Makinesi: *, /0, %



switch (op) → "%": hiçbir case eşleşmez → **default**.

Adım 9 / 10

```
1  double x = klavye.nextDouble();
2  String op = klavye.next();
3  double y = klavye.nextDouble();
▶  switch (op) {
5    case "+" -> System.out.println(x + y);
6    case "-" -> System.out.println(x - y);
7    case "*" -> System.out.println(x * y);
8    case "/" -> {
9      if (y == 0) System.out.println("Hata!");
10     else System.out.println(x / y);
11   }
12   default -> System.out.println("Geçersiz");
13 }
```

DEĞİŞKENLER

x = 5.0 | op = "%" | y = 3.0

ÇIKTI

ÖRNEK · KODU ADIM ADIM ÇALIŞTIR

switch ile Hesap Makinesi: *, /0, %



Geçersiz: % bu hesap makinesinde tanımlı değil; default beklenmeyen girdiyi yakalar.

Adım 10 / 10

```
1  double x = klavye.nextDouble();
2  String op = klavye.next();
3  double y = klavye.nextDouble();
4  switch (op) {
5    case "+" -> System.out.println(x + y);
6    case "-" -> System.out.println(x - y);
7    case "*" -> System.out.println(x * y);
8    case "/" -> {
9      if (y == 0) System.out.println("Hata!");
10     else System.out.println(x / y);
11   }
▶    default -> System.out.println("Geçersiz");
13 }
```

DEĞİŞKENLER

x = 5.0 | op = "%" | y = 3.0

ÇIKTI

```
Geçersiz
```

ÖZET

Bölüm Özeti

- Sıralı yapı deyimleri yazıldığı sırada yürütür; karar yapısı koşula göre yol seçer.
- Koşul her zaman boolean bir değer üretmelidir.
- else-if zincirinde koşullar yukarıdan aşağı denenir; ilk doğru dal çalışır.
- Aralık ve karmaşık koşullarda if-else, tek değerin ayrık seçeneklerinde switch uygundur.

ÖZET

Kontrol Soruları

1. Bağımsız if yapıları ile else-if zinciri nasıl farklıdır?
2. Not harf dönüşümünde eşikler neden büyükten küçüğe yazılır?
3. İç içe karar yerine && ne zaman kullanılabilir?
4. switch için default neden yararlıdır?
5. Bir sayının çift olduğu hangi koşulla anlaşılır?

Alıştırmalar

1. Girilen tam sayının pozitif, negatif veya sıfır olduğunu yazdırın.
2. Boy ve kilodan vücut kitle indeksini (kilo / boy^2) hesaplayıp zayıf / normal / fazla kilolu sınıfını yazdırın.
3. Üç kenar uzunluğu bir üçgen oluşturuyor mu? Oluşturuyorsa eşkenar, ikizkenar veya çeşitkenar mı?
4. 1–7 arası gün numarasını switch ile gün adına çevirin; geçersiz girdide uyarı verin.
5. İki sayının küçüğünü `?:` ile bulun; ardından üç sayının küçüğüne genişletin.
6. Harf notu programına "not 0–100 dışında" kontrolü ekleyin. Bu kontrol zincirin neresine yazılmalı?

ÖZET

Kaynaklar

- Deitel ve Deitel, Java: How to Program, Early Objects, 11. baskı, Pearson, 2018.
- Liang, Introduction to Java Programming and Data Structures, 13. baskı, Pearson, 2024.
- Oracle, The Java Tutorials: **The if-then and if-then-else Statements**, **The switch Statement**.
- Oracle, Java Language Specification SE 21: **Bölüm 14, Blocks, Statements, and Patterns**.
- Sedgewick ve Wayne, Computer Science: An Interdisciplinary Approach, Addison-Wesley, 2016: **1.3 Conditionals and Loops**.
- Downey ve Mayfield, Think Java, 2. baskı, O'Reilly, 2019 (**ücretsiz çevrim içi sürüm**).

Son

Koşullu İfadeler

Sonraki sunum: **Döngüler** · **Çalışma soruları** · Simülatör: **Karar İzleyici** · **Terim sözlüğü**