Bölüm 8: Döngüler

Bilgisayar ve Programlamaya Giriş

for · while · do-while · İç içe · Örnekler

Sercan KÜLCÜ · Giresun Üniversitesi

© Sercan Külcü. Tüm hakları saklıdır. · Sunumun etkileşimli HTML sürümü: https://sercankulcu.github.io/files/computer/slides/Bolum_08_Donguler.html

GİRİŞ

Öğrenme Çıktıları

Döngü

for, while ve do-while yapılarını seçmek.

Kalıplar

Sayaç ve biriktirici ile hesap yapmak.

İç içe

İç içe döngülerle satır-sütun desenleri üretmek.

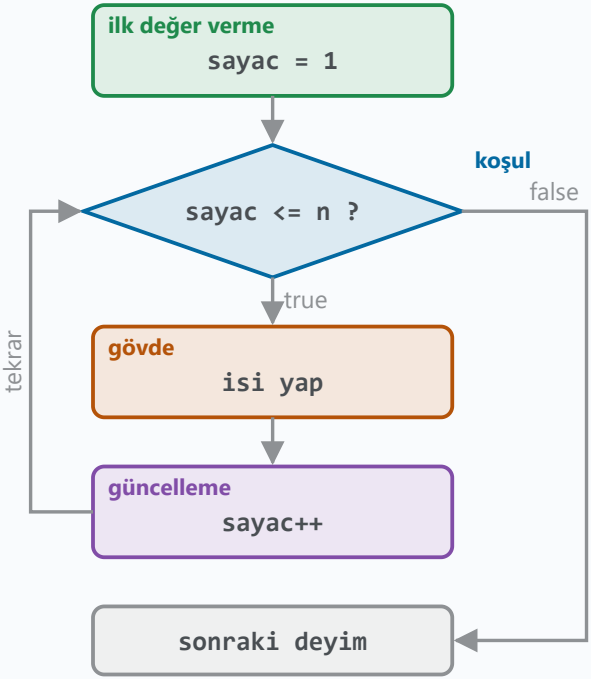
İzleme

Döngüyü sınır değerlerle adım adım doğrulamak.

Döngünün Bileşenleri

- **İlk değer verme**: döngüden önce bir kez kurulan durum.
- **Koşul**: her turdan önce (do-while'da sonra) sorulur; false ise çıkılır.
- **Gövde**: her turda yapılan iş.
- **Güncelleme**: koşulu sona yaklaştırır; unutulursa döngü bitmez.

Bu dört renk bölüm boyunca aynı anlamda kullanılır.

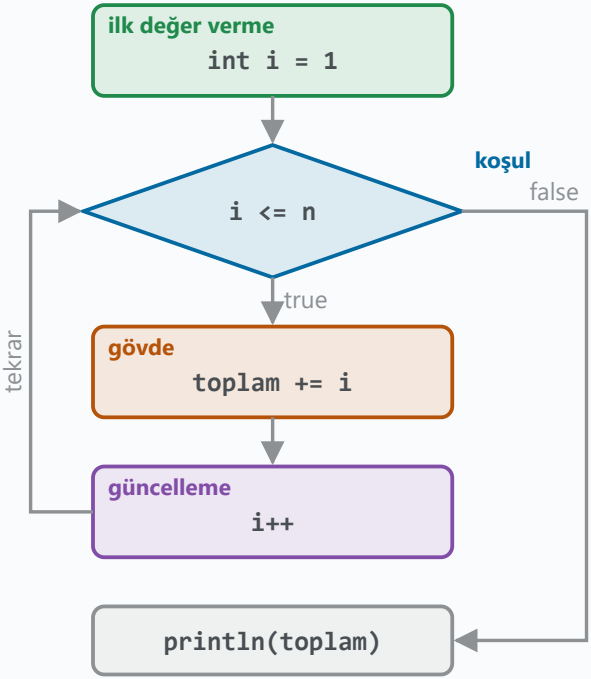


for Döngüsü

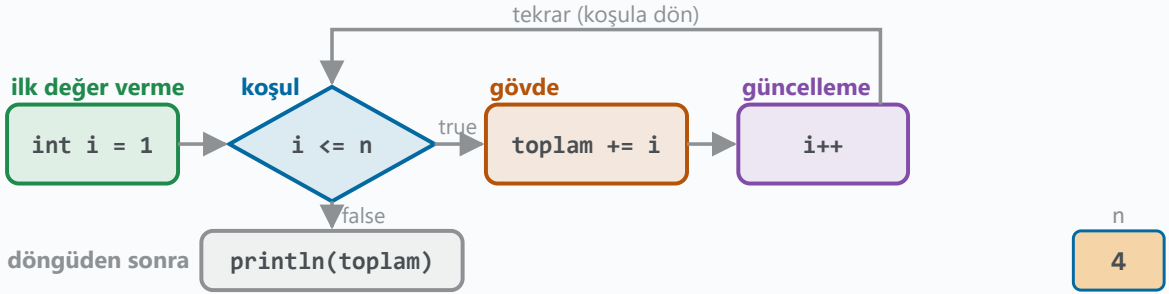
```
int n = 4;
int toplam = 0;
for (int i = 1; i <= n; i++) {
    toplam += i;
}
System.out.println(toplam);
```

ilk değer verme koşul gövde güncelleme

- Tekrar sayısı belli olduğunda for uygundur; dört parça tek satırda toplanır.
- Sıra: ilk değer verme → koşul → gövde → güncelleme → koşul …
- i yalnızca döngü içinde tanımlıdır.



Kod İzleme: `for` ile 1 + 2 + 3 + 4



n = 4: 1'den 4'e kadar toplayacağız.

Adım 1 / 17

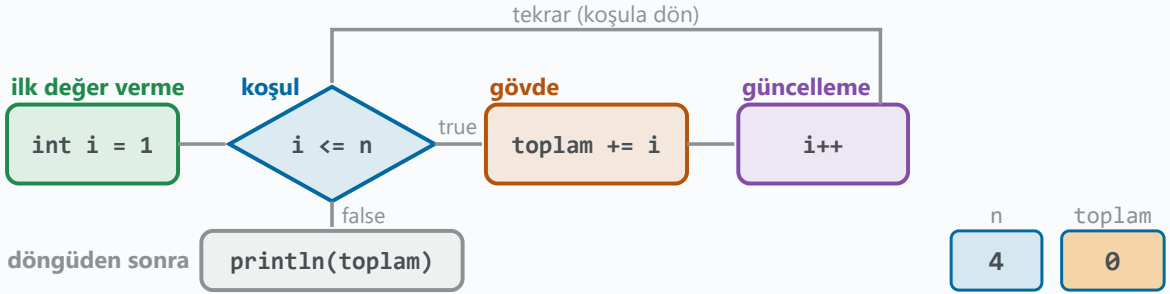
```
▶ int n = 4;
2 int toplam = 0;
3 for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

n = 4

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Biriktirici **toplam = 0** ile başlar (henüz hiçbir sayı eklenmedi).

Adım 2 / 17

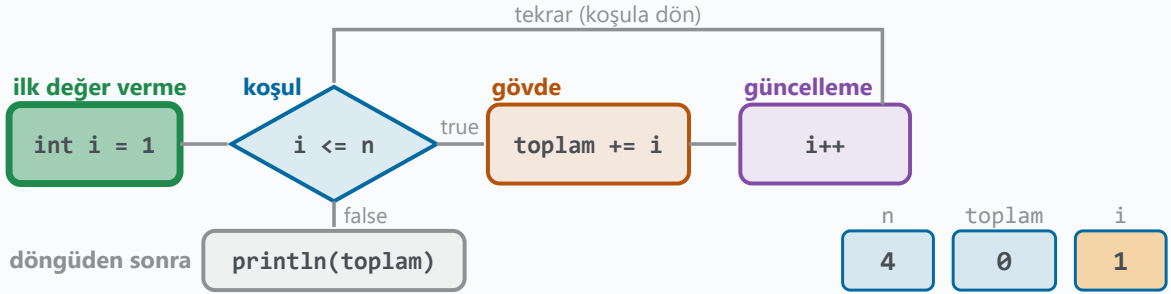
```
1 int n = 4;
▶ int toplam = 0;
3 for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 0`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



İlk değer verme: `int i = 1` yalnızca bir kez çalışır.

Adım 3 / 17

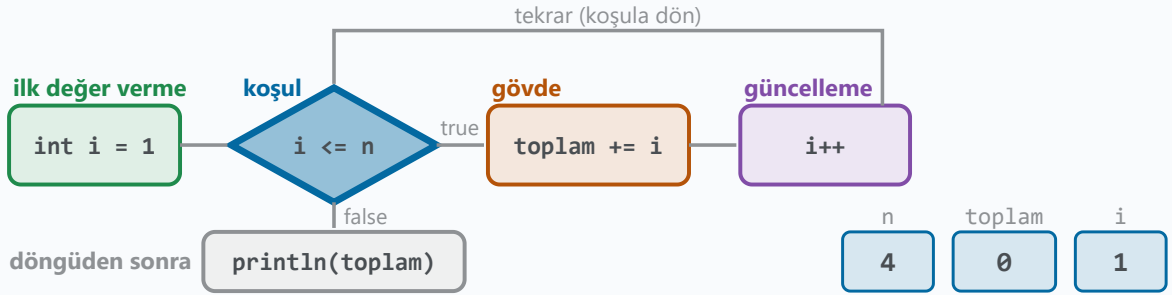
```
1 int n = 4;
2 int toplam = 0;
▶ for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 0` `i = 1`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Koşul: `i <= n` → 1 <= 4 → **true** → gövdeye gir.

Adım 4 / 17

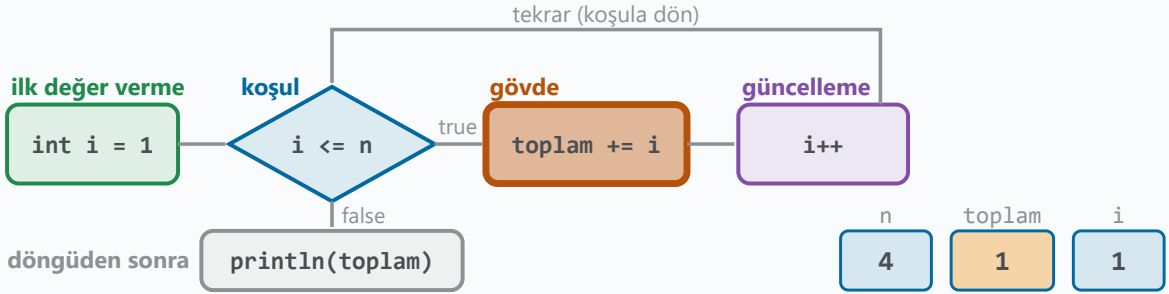
```
1 int n = 4;
2 int toplam = 0;
▶ for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 0` `i = 1`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Gövde: toplam = 0 + 1 = **1**.

Adım 5 / 17

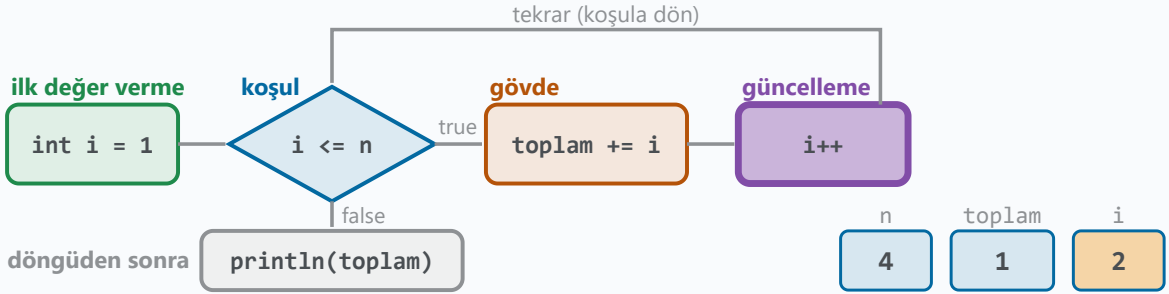
```
1 int n = 4;
2 int toplam = 0;
3 for (int i = 1; i <= n; i++) {
▶     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 1` `i = 1`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Güncelleme: `i++` → i = 2. Sonra yine koşula dönülür.

Adım 6 / 17

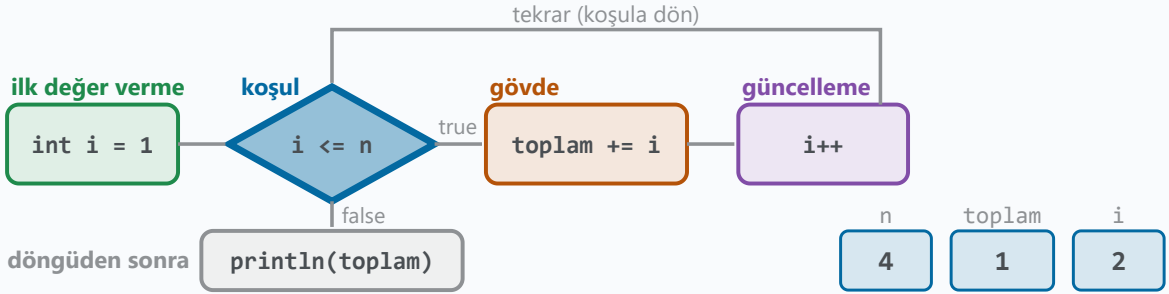
```
1 int n = 4;
2 int toplam = 0;
▶ for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 1` `i = 2`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Koşul: `i <= n` → 2 <= 4 → **true** → gövdeye gir.

Adım 7 / 17

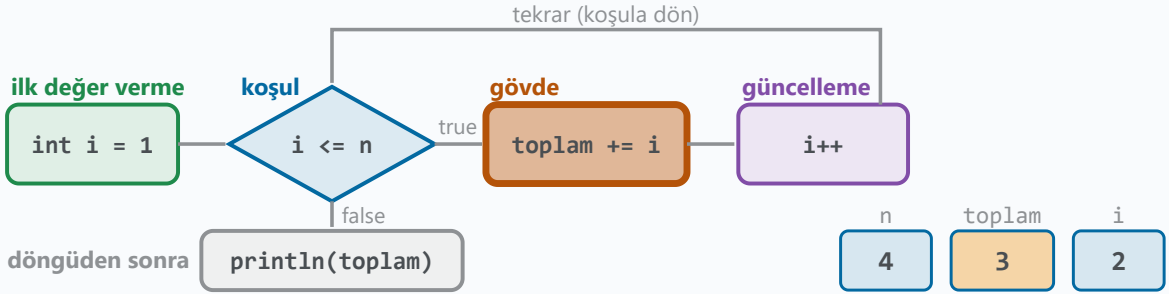
```
1 int n = 4;
2 int toplam = 0;
▶ for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 1` `i = 2`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Gövde: toplam = 1 + 2 = **3**.

Adım 8 / 17

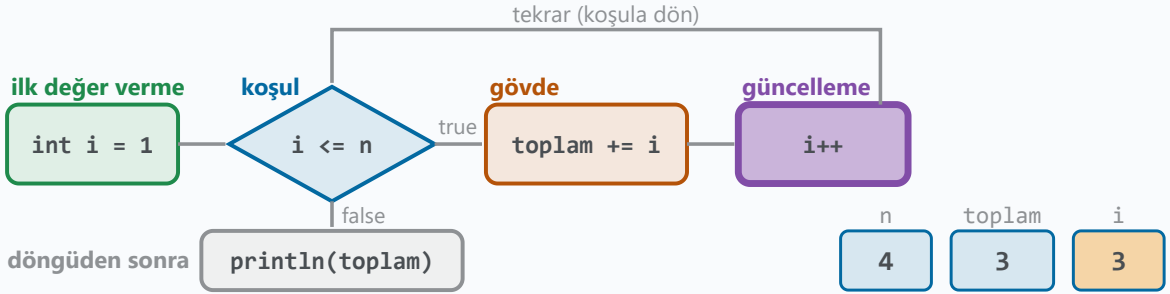
```
1 int n = 4;
2 int toplam = 0;
3 for (int i = 1; i <= n; i++) {
▶     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 3` `i = 2`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Güncelleme: `i++` → i = 3. Sonra yine koşula dönülür.

Adım 9 / 17

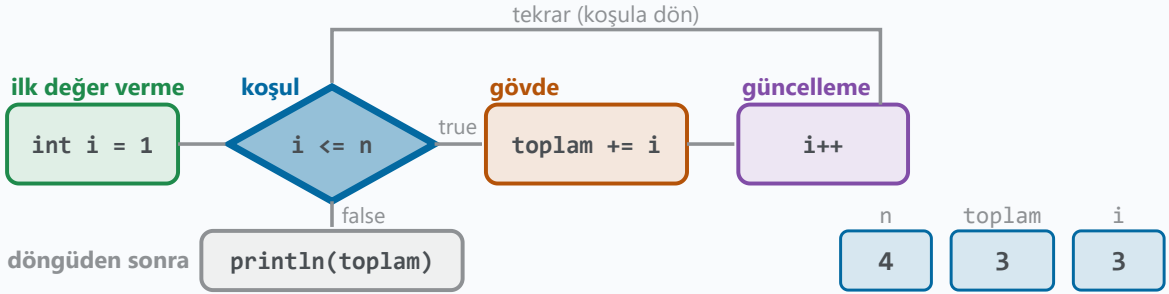
```
1 int n = 4;
2 int toplam = 0;
▶ for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 3` `i = 3`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Koşul: `i <= n` → 3 <= 4 → **true** → gövdeye gir.

Adım 10 / 17

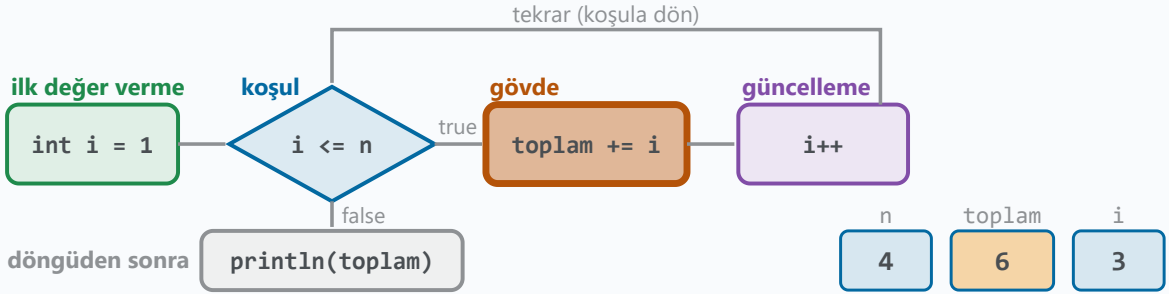
```
1 int n = 4;
2 int toplam = 0;
▶ for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 3` `i = 3`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Gövde: toplam = 3 + 3 = **6**.

Adım 11 / 17

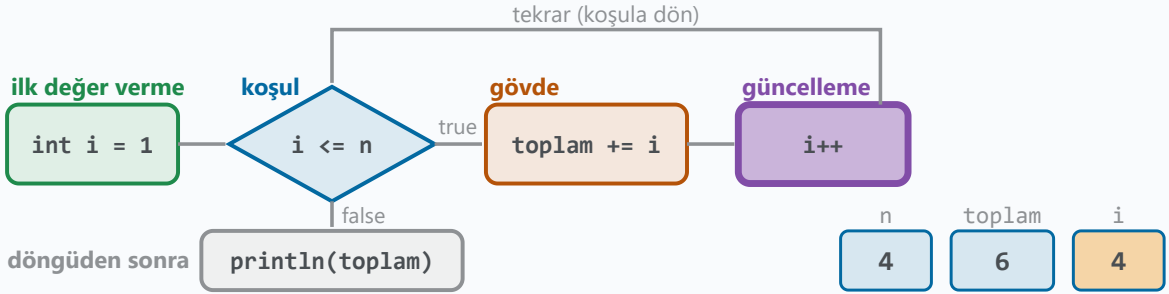
```
1 int n = 4;
2 int toplam = 0;
3 for (int i = 1; i <= n; i++) {
▶     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 6` `i = 3`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Güncelleme: `i++` → i = 4. Sonra yine koşula dönülür.

Adım 12 / 17

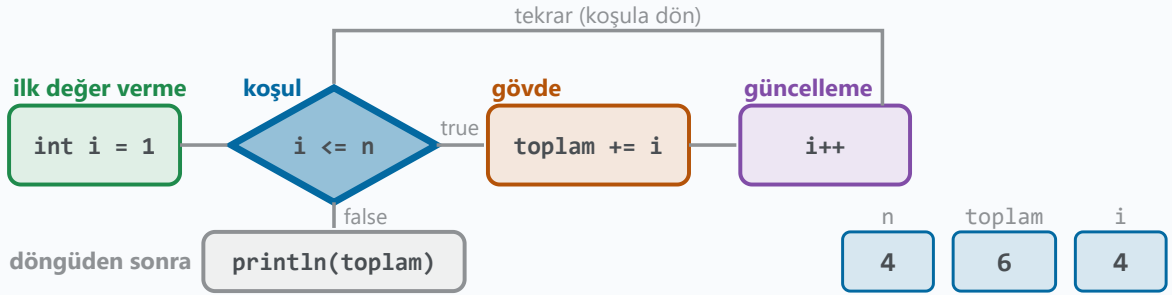
```
1 int n = 4;
2 int toplam = 0;
▶ for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 6` `i = 4`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Koşul: `i <= n` → 4 <= 4 → **true** → gövdeye gir.

Adım 13 / 17

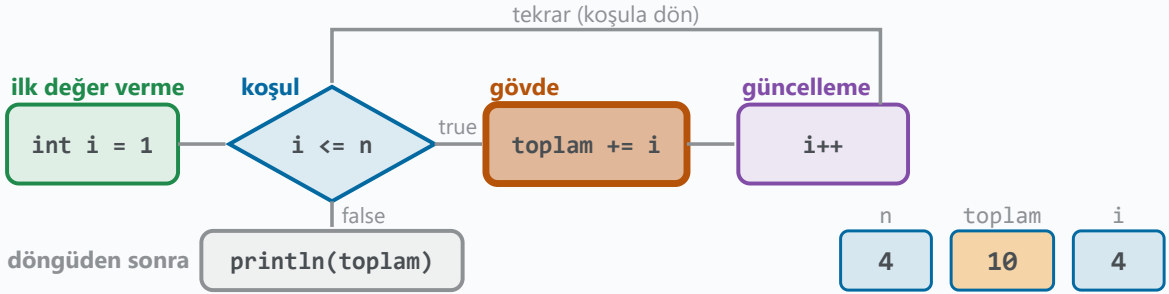
```
1 int n = 4;
2 int toplam = 0;
▶ for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 6` `i = 4`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Gövde: toplam = 6 + 4 = **10**.

Adım 14 / 17

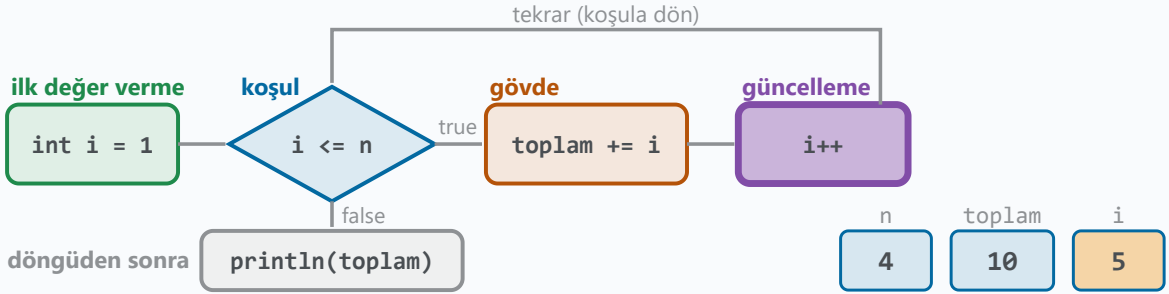
```
1 int n = 4;
2 int toplam = 0;
3 for (int i = 1; i <= n; i++) {
▶     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 10` `i = 4`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Güncelleme: `i++` → i = 5. Sonra yine koşula dönülür.

Adım 15 / 17

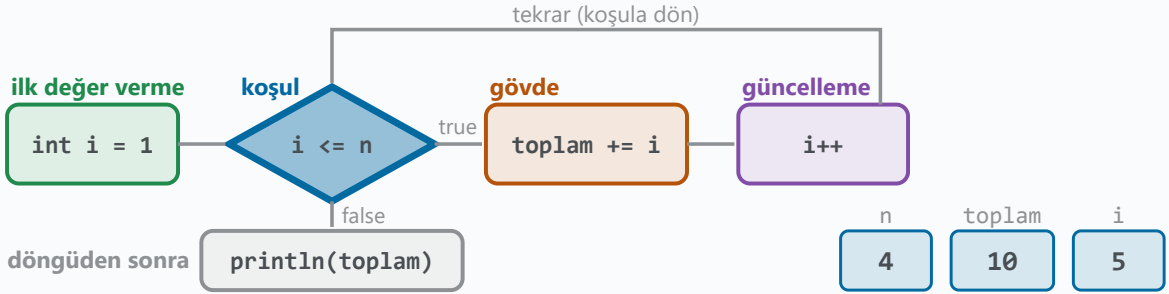
```
1 int n = 4;
2 int toplam = 0;
▶ for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 10` `i = 5`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Koşul: `i <= n` → 5 <= 4 → **false** → döngü biter.

Adım 16 / 17

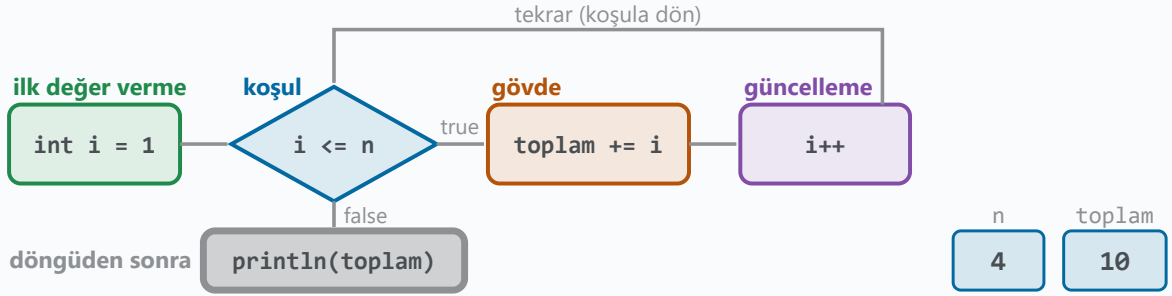
```
1 int n = 4;
2 int toplam = 0;
▶ for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
6 System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 10` `i = 5`

ÇIKTI

Kod İzleme: `for` ile 1 + 2 + 3 + 4



Döngüden sonra `println` **10** yazar. i artık yok: kapsamı döngüydü. Gövde 4 kez, koşul 5 kez çalıştı.

Adım 17 / 17

```
1 int n = 4;
2 int toplam = 0;
3 for (int i = 1; i <= n; i++) {
4     toplam += i;
5 }
▶ System.out.println(toplam);
```

DEĞİŞKENLER

`n = 4` `toplam = 10`

ÇIKTI

```
10
```

for Döngüsünün Esnek Biçimleri

Başlık	Çalışma
`for (int i = 0, j = 9; i < j; i++, j--)`	i ve j birlikte: (0,9) … (4,5), 5 tur
`for (int i = 0; i <= 10; i += 2)`	0 2 4 6 8 10
`for (int i = 5; i >= 1; i--)`	geri sayım: 5 4 3 2 1

```
int k = 0;
for (;;) {            // üç bölüm de boş: koşul true sayılır
    k++;
    if (k == 3) break;  // çıkış koşulu şart
}
System.out.println(k);   // 3
```

- İlk değer verme, koşul ve güncelleme bölümlerinin her biri isteğe bağlıdır.
- İlk değer verme bölümünde aynı tipte birden çok değişken, güncellemede virgülle birden çok ifade yazılabilir.

`++i` ile `i++` Farkı

```
for (int i = 0; i < 3; i++) System.out.print(i + " ");   // 0 1 2
for (int i = 0; i < 3; ++i) System.out.print(i + " ");   // 0 1 2
```

for'un güncelleme bölümünde ifadenin değeri kullanılmaz; ikisi de yalnızca i'yi 1 artırır, **fark yoktur**.

Başlangıç	İfade	sonuç	sonra
`a = 5`	`b = a++`	b = 5	a = 6
`a = 5`	`b = ++a`	b = 6	a = 6
`idx = 0`	`d[idx++]`	10	idx = 1
`idx = 0`	`d[++idx]`	20	idx = 1

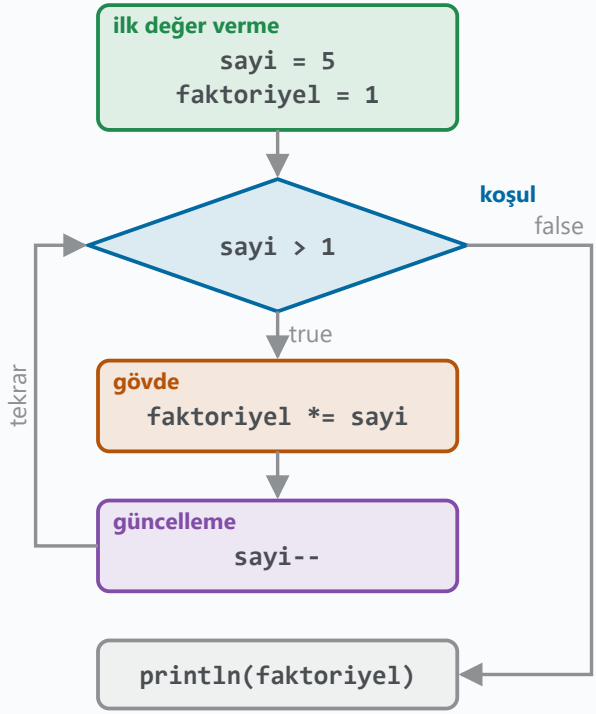
- `i++` önce eski değeri verir, sonra artırır; `++i` önce artırır, sonra yeni değeri verir.
- Tabloda `d = {10, 20, 30}` ; fark yalnızca değer kullanıldığında ortaya çıkar. Okunabilirlik için sayaçta `i++` yazın.

while Döngüsü

```
int sayi = 5;
int faktoriyel = 1;
while (sayi > 1) {
    faktoriyel *= sayi;
    sayi--;
}
System.out.println(faktoriyel);
```

ilk değer verme koşul gövde güncelleme

- Koşul gövdeden **önce** kontrol edilir.
- Başlangıçta false ise gövde hiç çalışmaz.
- İlk değer verme döngüden önce, güncelleme gövdenin içinde yazılır.



Kod İzleme: `while` ile 5!



İlk değer verme: sayi = 5 (5! hesaplanacak).

Adım 1 / 16

```
▶ int sayi = 5;
2 int faktoriyel = 1;
3 while (sayi > 1) {
4     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 5`

ÇIKTI

Kod İzleme: `while` ile 5!



Çarpım biriktiricisi **1** ile başlar (0 olsaydı sonuç hep 0 olurdu).

Adım 2 / 16

```
1 int sayi = 5;
▶ int faktoriyel = 1;
3 while (sayi > 1) {
4     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 5` `faktoriyel = 1`

ÇIKTI

Kod İzleme: `while` ile 5!



Koşul: 5 > 1 → **true** → gövde çalışır.

Adım 3 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
▶ while (sayi > 1) {
4     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 5` `faktoriyel = 1`

ÇIKTI

Kod İzleme: `while` ile 5!



faktoriyel = 1 × 5 = **5**.

Adım 4 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
3 while (sayi > 1) {
▶     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 5` `faktoriyel = 5`

ÇIKTI

Kod İzleme: `while` ile 5!



Güncelleme: sayi-- → **4**. Koşul sona yaklaşır.

Adım 5 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
3 while (sayi > 1) {
4     faktoriyel *= sayi;
▶     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 4` `faktoriyel = 5`

ÇIKTI

Kod İzleme: `while` ile 5!



Koşul: 4 > 1 → **true** → gövde çalışır.

Adım 6 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
▶ while (sayi > 1) {
4     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 4` `faktoriyel = 5`

ÇIKTI

Kod İzleme: `while` ile 5!



faktoriyel = 5 × 4 = **20**.

Adım 7 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
3 while (sayi > 1) {
▶     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 4` `faktoriyel = 20`

ÇIKTI

Kod İzleme: `while` ile 5!



Güncelleme: sayi-- → **3**. Koşul sona yaklaşır.

Adım 8 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
3 while (sayi > 1) {
4     faktoriyel *= sayi;
▶     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 3` `faktoriyel = 20`

ÇIKTI

Kod İzleme: `while` ile 5!



Koşul: 3 > 1 → **true** → gövde çalışır.

Adım 9 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
▶ while (sayi > 1) {
4     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 3` `faktoriyel = 20`

ÇIKTI

Kod İzleme: `while` ile 5!



faktoriyel = 20 × 3 = **60**.

Adım 10 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
3 while (sayi > 1) {
▶     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 3` `faktoriyel = 60`

ÇIKTI

Kod İzleme: `while` ile 5!



Güncelleme: sayi-- → **2**. Koşul sona yaklaşır.

Adım 11 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
3 while (sayi > 1) {
4     faktoriyel *= sayi;
▶     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 2` `faktoriyel = 60`

ÇIKTI

Kod İzleme: `while` ile 5!



Koşul: 2 > 1 → **true** → gövde çalışır.

Adım 12 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
▶ while (sayi > 1) {
4     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 2` `faktoriyel = 60`

ÇIKTI

Kod İzleme: `while` ile 5!



faktoriyel = 60 × 2 = **120**.

Adım 13 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
3 while (sayi > 1) {
▶     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 2` `faktoriyel = 120`

ÇIKTI

Kod İzleme: `while` ile 5!



Güncelleme: sayi-- → **1**. Koşul sona yaklaşır.

Adım 14 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
3 while (sayi > 1) {
4     faktoriyel *= sayi;
▶     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 1` `faktoriyel = 120`

ÇIKTI

Kod İzleme: `while` ile 5!



Koşul: 1 > 1 → **false** → döngüden çıkılır.

Adım 15 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
▶ while (sayi > 1) {
4     faktoriyel *= sayi;
5     sayi--;
6 }
7 System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 1` `faktoriyel = 120`

ÇIKTI

Kod İzleme: `while` ile 5!



120 yazılır: 5 × 4 × 3 × 2 = 120. Gövde 4 kez çalıştı; 1 ile çarpmaya gerek yok.

Adım 16 / 16

```
1 int sayi = 5;
2 int faktoriyel = 1;
3 while (sayi > 1) {
4     faktoriyel *= sayi;
5     sayi--;
6 }
▶ System.out.println(faktoriyel);
```

DEĞİŞKENLER

`sayi = 1` `faktoriyel = 120`

ÇIKTI

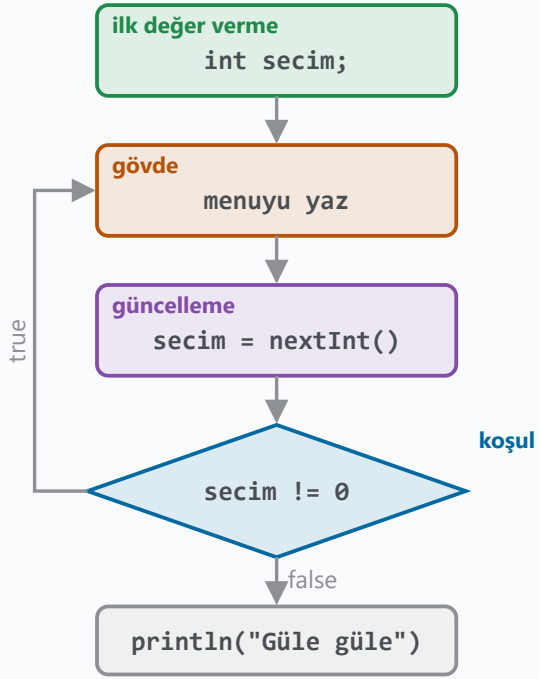
```
120
```

do-while Döngüsü

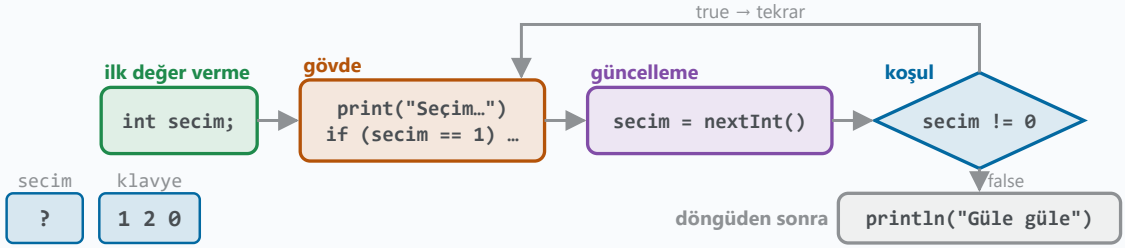
```
Scanner klavye = new Scanner(System.in);
int secim;
do {
   System.out.print("Seçim (1/0): ");
   secim = klavye.nextInt();
   if (secim == 1)
      System.out.println("Merhaba!");
} while (secim != 0);
System.out.println("Güle güle");
```

ilk değer verme koşul gövde güncelleme

- Koşul gövdeden **sonra** kontrol edilir → gövde en az bir kez çalışır.
- Menü ve girdi doğrulama için doğal seçimdir.
- `while (…);` sonundaki noktalı virgül unutulmamalı.



Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



Klavyeden okumak için Scanner (dosya başında `import java.util.Scanner;`). Kullanıcı sırayla **1, 2, 0** yazacak.

Adım 1 / 15

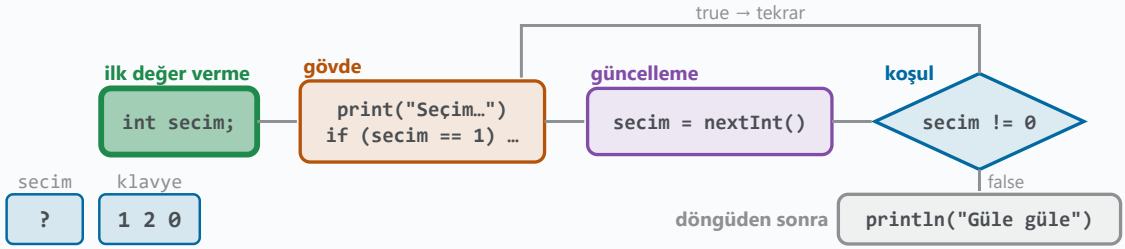
```
▶ Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

DEĞİŞKENLER

`klavye = Scanner`

ÇIKTI

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



secim bildirildi ama değeri yok; koşul ilk kez gövdeden sonra sorulacağı için sorun olmaz.

Adım 2 / 15

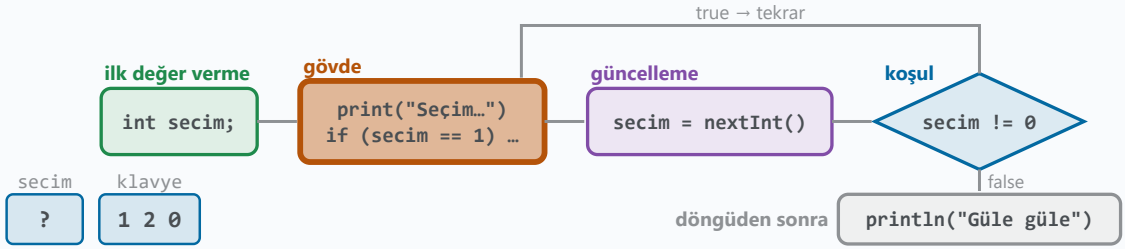
```
1 Scanner klavye = new Scanner(System.in);
▶ int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

DEĞİŞKENLER

`klavye = Scanner` `secim = ?`

ÇIKTI

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



1. tur: **koşula bakmadan** gövdeye girilir, menü yazılır.

Adım 3 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
▶   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

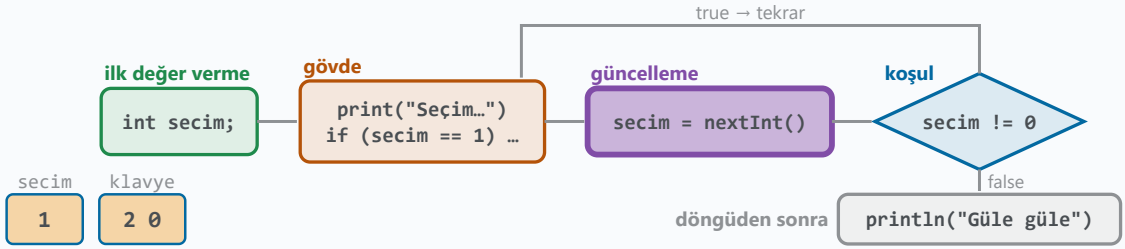
DEĞİŞKENLER

`klavye = Scanner` `secim = ?`

ÇIKTI

```
Seçim (1/0):
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



Kullanıcı **1** yazar → secim = 1.

Adım 4 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
▶   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

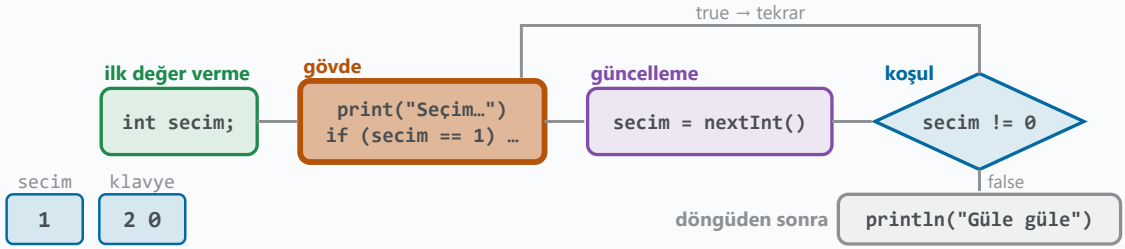
DEĞİŞKENLER

`klavye = Scanner` `secim = 1`

ÇIKTI

```
Seçim (1/0): 1
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



secim == 1 → true: «Merhaba!» yazılır.

Adım 5 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
▶  if (secim == 1)
▶    System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

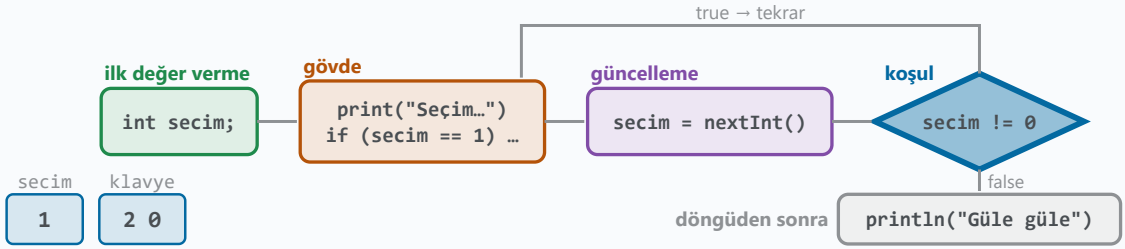
DEĞİŞKENLER

`klavye = Scanner` `secim = 1`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



Koşul: 1 != 0 → **true** → başa dön.

Adım 6 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
▶ } while (secim != 0);
9 System.out.println("Güle güle");
```

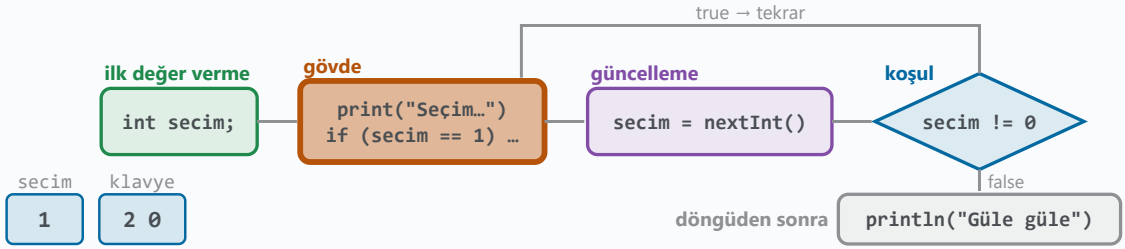
DEĞİŞKENLER

`klavye = Scanner` `secim = 1`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



2. tur: **koşula bakmadan** gövdeye girilir, menü yazılır.

Adım 7 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
▶   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

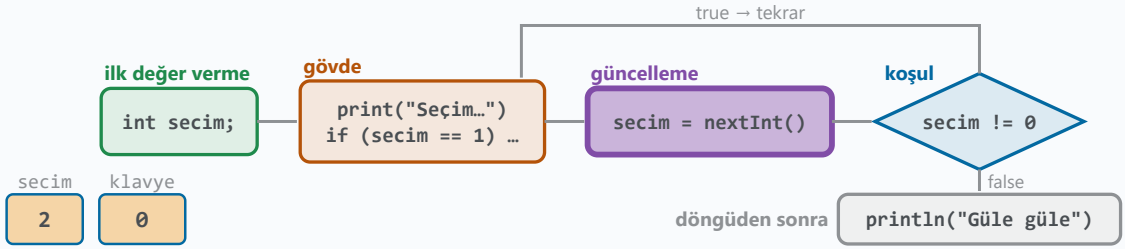
DEĞİŞKENLER

`klavye = Scanner` `secim = 1`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
Seçim (1/0):
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



Kullanıcı **2** yazar → secim = 2.

Adım 8 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
▶   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

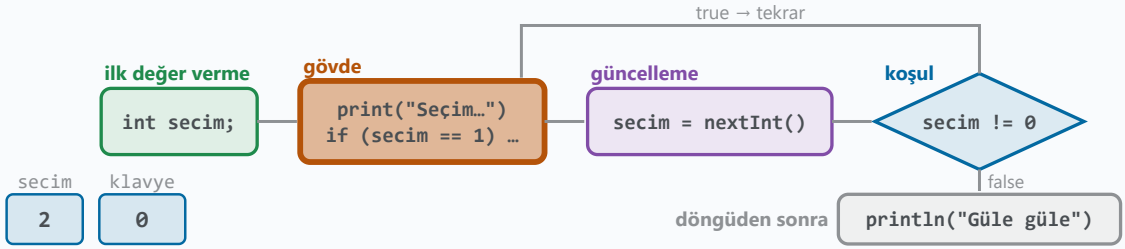
DEĞİŞKENLER

`klavye = Scanner` `secim = 2`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
Seçim (1/0): 2
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



secim == 1 → false: hiçbir şey yapılmaz.

Adım 9 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
▶  if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

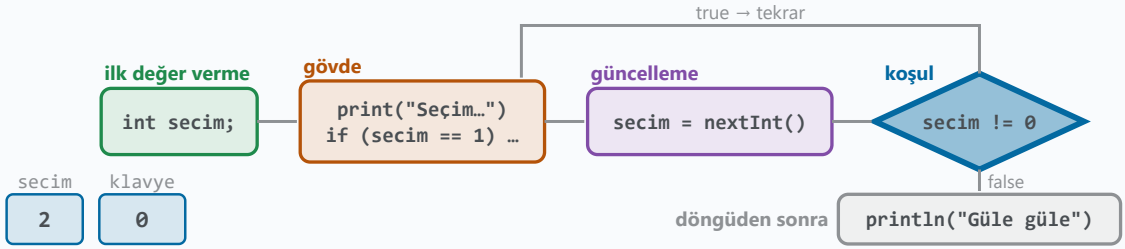
DEĞİŞKENLER

`klavye = Scanner` `secim = 2`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
Seçim (1/0): 2
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



Koşul: 2 != 0 → **true** → başa dön.

Adım 10 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
▶ } while (secim != 0);
9 System.out.println("Güle güle");
```

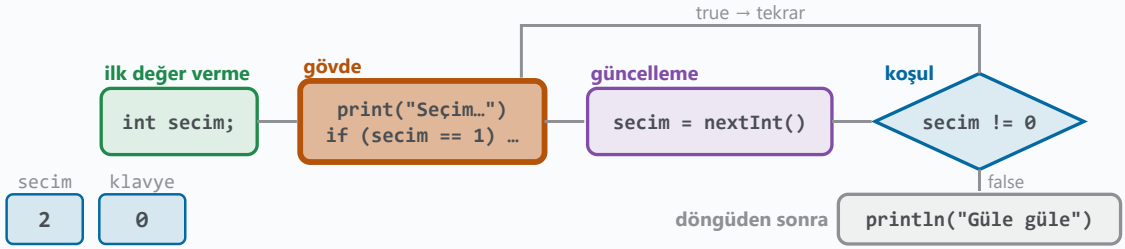
DEĞİŞKENLER

`klavye = Scanner` `secim = 2`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
Seçim (1/0): 2
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



3. tur: **koşula bakmadan** gövdeye girilir, menü yazılır.

Adım 11 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
▶   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

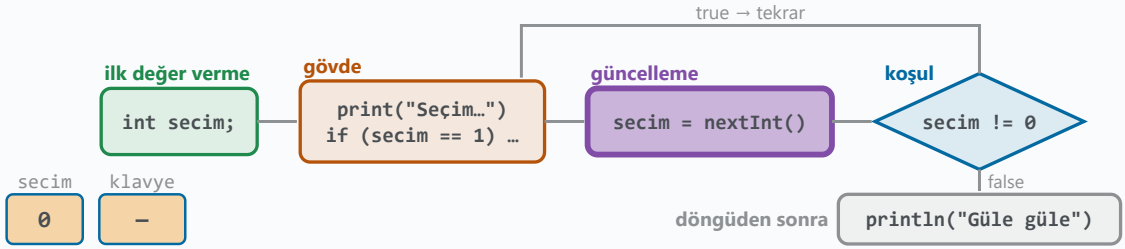
DEĞİŞKENLER

`klavye = Scanner` `secim = 2`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
Seçim (1/0): 2
Seçim (1/0):
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



Kullanıcı **0** yazar → secim = 0.

Adım 12 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
▶   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

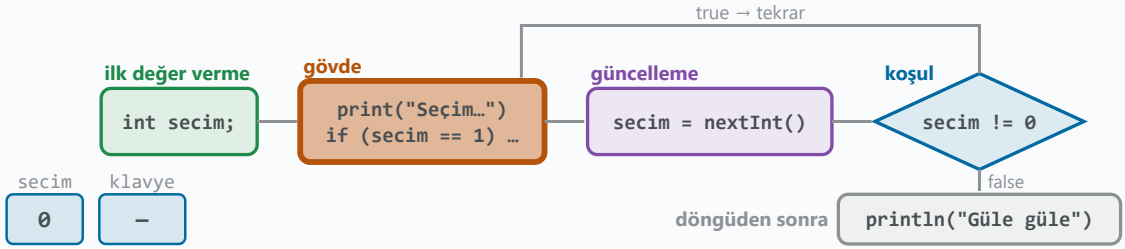
DEĞİŞKENLER

`klavye = Scanner` `secim = 0`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
Seçim (1/0): 2
Seçim (1/0): 0
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



secim == 1 → false: hiçbir şey yapılmaz.

Adım 13 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
▶  if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
9 System.out.println("Güle güle");
```

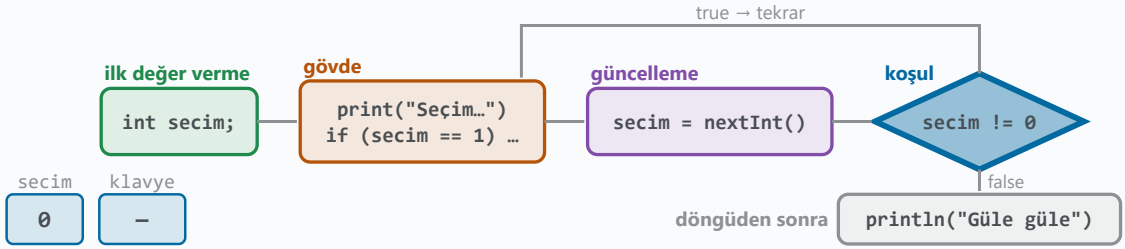
DEĞİŞKENLER

`klavye = Scanner` `secim = 0`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
Seçim (1/0): 2
Seçim (1/0): 0
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



Koşul: 0 != 0 → **false** → döngü biter.


```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
▶ } while (secim != 0);
9 System.out.println("Güle güle");
```

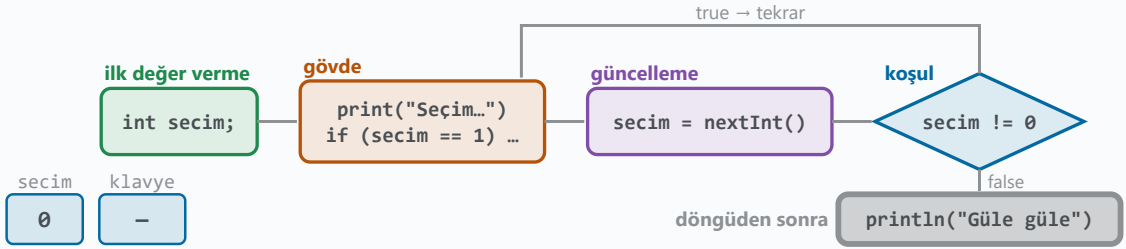
DEĞİŞKENLER

`klavye = Scanner` `secim = 0`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
Seçim (1/0): 2
Seçim (1/0): 0
```

Kod İzleme: `do-while` menüsü (girdi 1, 2, 0)



Gövde 3 kez çalıştı; koşul da 3 kez soruldu (while'da gövde sayısından bir fazla sorulurdu).

Adım 15 / 15

```
1 Scanner klavye = new Scanner(System.in);
2 int secim;
3 do {
4   System.out.print("Seçim (1/0): ");
5   secim = klavye.nextInt();
6   if (secim == 1)
7     System.out.println("Merhaba!");
8 } while (secim != 0);
▶ System.out.println("Güle güle");
```

DEĞİŞKENLER

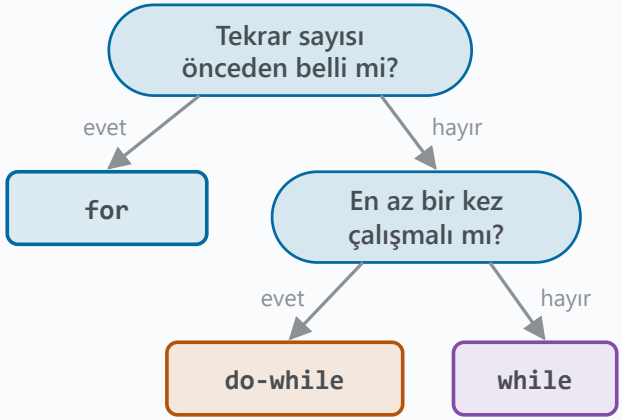
`klavye = Scanner` `secim = 0`

ÇIKTI

```
Seçim (1/0): 1
Merhaba!
Seçim (1/0): 2
Seçim (1/0): 0
Güle güle
```

Döngü Seçimi

Durum	Tercih
1’den n’ye say, n kez tekrarla	for
Sayı 0 olana kadar basamak işle	while
Menü göster, geçerli girdi gelene kadar sor	do-while
Kullanıcı “0” yazana kadar topla	while / do-while

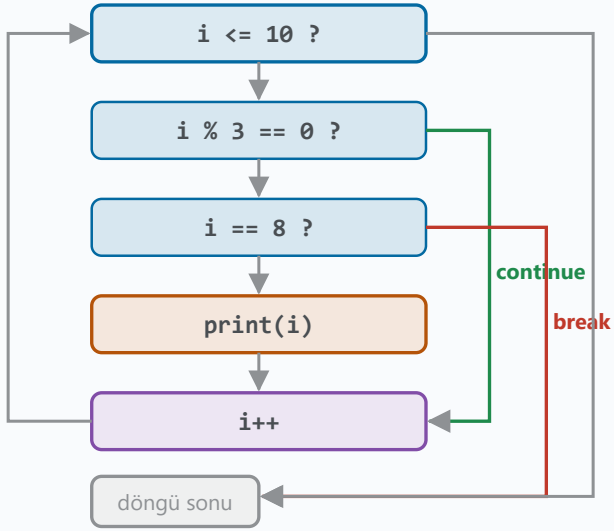


Üçü de birbirine çevrilebilir; seçim okunabilirlik içindir.

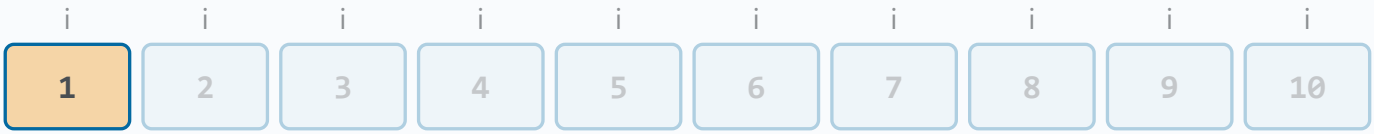
break ve continue

```
for (int i = 1; i <= 10; i++) {
    if (i % 3 == 0) continue;
    if (i == 8) break;
    System.out.print(i + " ");
}
System.out.println("| bitti");
```

- **continue**: turun kalanını atlar, **güncellemeye** (i++) geçer.
- **break**: en yakın döngüyü hemen bitirir.
- Aşırı kullanım akışı izlemeyi zorlaştırır.



Kod İzleme: `break` ve `continue`



i = 1: 1 % 3 = 1 → false; 1 == 8 → false. İki kontrol de geçildi.

Adım 1 / 14

```
1 for (int i = 1; i <= 10; i++) {
▶     if (i % 3 == 0) continue;
▶     if (i == 8) break;
4     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

DEĞİŞKENLER

`i = 1`

ÇIKTI

Kod İzleme: `break` ve `continue`



1 yazılır.

Adım 2 / 14

```
1 for (int i = 1; i <= 10; i++) {
2     if (i % 3 == 0) continue;
3     if (i == 8) break;
▶     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

DEĞİŞKENLER

`i = 1`

ÇIKTI

```
1
```

Kod İzleme: `break` ve `continue`



i = 2: 2 % 3 = 2 → false; 2 == 8 → false. İki kontrol de geçildi.

Adım 3 / 14

```
1 for (int i = 1; i <= 10; i++) {
▶     if (i % 3 == 0) continue;
▶     if (i == 8) break;
4     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

DEĞİŞKENLER

`i = 2`

ÇIKTI

```
1
```

Kod İzleme: `break` ve `continue`



2 yazılır.

Adım 4 / 14

```
1 for (int i = 1; i <= 10; i++) {
2     if (i % 3 == 0) continue;
3     if (i == 8) break;
▶     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

DEĞİŞKENLER

`i = 2`

ÇIKTI

```
1 2
```

Kod İzleme: `break` ve `continue`



i = 3: 3 % 3 == 0 → **true** → **continue**: print atlanır, doğrudan i++.

Adım 5 / 14

```
1 for (int i = 1; i <= 10; i++) {
▶     if (i % 3 == 0) continue;
3     if (i == 8) break;
4     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

DEĞİŞKENLER

`i = 3`

ÇIKTI

```
1 2
```

Kod İzleme: `break` ve `continue`



i = 4: 4 % 3 = 1 → false; 4 == 8 → false. İki kontrol de geçildi.


```
1 for (int i = 1; i <= 10; i++) {
▶     if (i % 3 == 0) continue;
▶     if (i == 8) break;
4     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

DEĞİŞKENLER

`i = 4`

ÇIKTI

```
1 2
```

Kod İzleme: `break` ve `continue`



4 yazılır.

Adım 7 / 14

```
1 for (int i = 1; i <= 10; i++) {
2     if (i % 3 == 0) continue;
3     if (i == 8) break;
▶     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

DEĞİŞKENLER

`i = 4`

ÇIKTI

```
1 2 4
```

Kod İzleme: `break` ve `continue`



i = 5: 5 % 3 = 2 → false; 5 == 8 → false. İki kontrol de geçildi.

Adım 8 / 14

```
1 for (int i = 1; i <= 10; i++) {
▶     if (i % 3 == 0) continue;
▶     if (i == 8) break;
4     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

DEĞİŞKENLER

`i = 5`

ÇIKTI

```
1 2 4
```

Kod İzleme: `break` ve `continue`



5 yazılır.

Adım 9 / 14

```
1 for (int i = 1; i <= 10; i++) {
2     if (i % 3 == 0) continue;
3     if (i == 8) break;
▶     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

DEĞİŞKENLER

`i = 5`

ÇIKTI

```
1 2 4 5
```

Kod İzleme: `break` ve `continue`



i = 6: 6 % 3 == 0 → **true** → **continue**: print atlanır, doğrudan i++.

Adım 10 / 14

```
1 for (int i = 1; i <= 10; i++) {
▶       if (i % 3 == 0) continue;
3       if (i == 8) break;
4       System.out.print(i + " ");
5   }
6   System.out.println("| bitti");
```

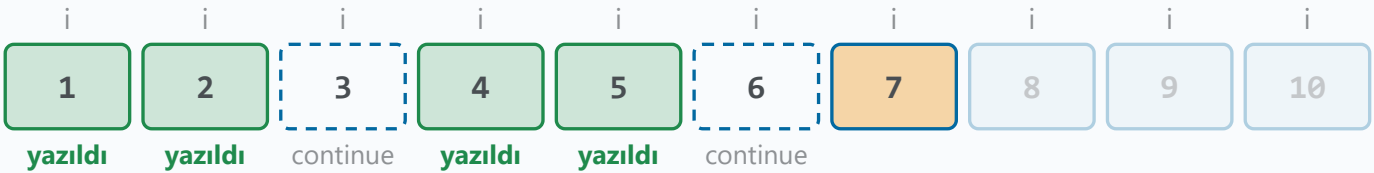
DEĞİŞKENLER

`i = 6`

ÇIKTI

```
1 2 4 5
```

Kod İzleme: `break` ve `continue`



i = 7: 7 % 3 = 1 → false; 7 == 8 → false. İki kontrol de geçildi.


```
1 for (int i = 1; i <= 10; i++) {
▶     if (i % 3 == 0) continue;
▶     if (i == 8) break;
4     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

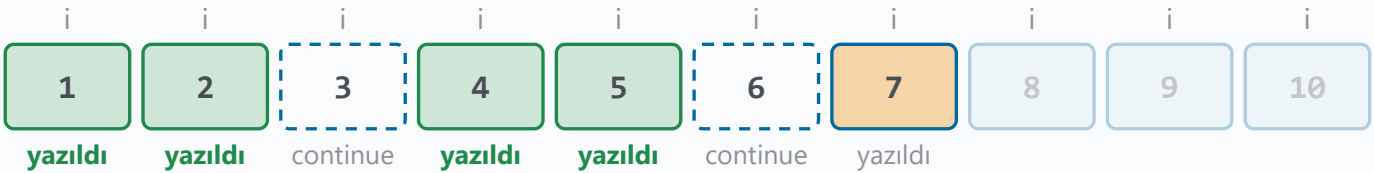
DEĞİŞKENLER

`i = 7`

ÇIKTI

```
1 2 4 5
```

Kod İzleme: `break` ve `continue`



7 yazılır.

Adım 12 / 14

```
1 for (int i = 1; i <= 10; i++) {
2     if (i % 3 == 0) continue;
3     if (i == 8) break;
▶     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

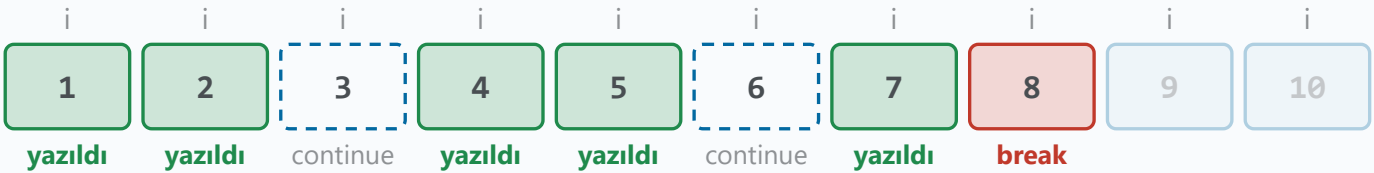
DEĞİŞKENLER

`i = 7`

ÇIKTI

```
1 2 4 5 7
```

Kod İzleme: `break` ve `continue`



i = 8: 8 % 3 = 2 → false; 8 == 8 → **true** → **break**: döngü burada biter, 9 ve 10'a hiç gelinmez.

Adım 13 / 14

```
1 for (int i = 1; i <= 10; i++) {
▶     if (i % 3 == 0) continue;
▶     if (i == 8) break;
4     System.out.print(i + " ");
5 }
6 System.out.println("| bitti");
```

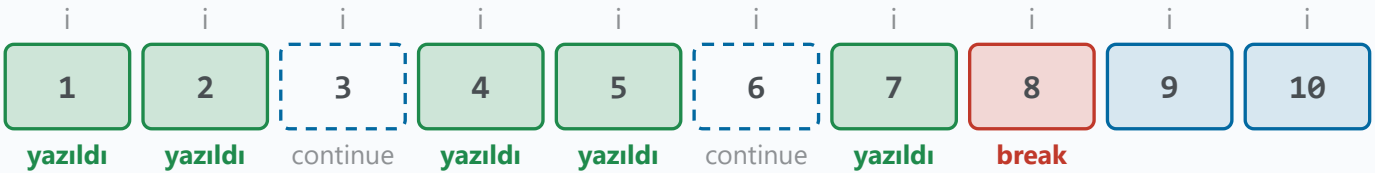
DEĞİŞKENLER

`i = 8`

ÇIKTI

```
1 2 4 5 7
```

Kod İzleme: `break` ve `continue`



Döngü break ile bitti; akış döngüden sonraki satırdan sürer.

Adım 14 / 14

```
1 for (int i = 1; i <= 10; i++) {
2     if (i % 3 == 0) continue;
3     if (i == 8) break;
4     System.out.print(i + " ");
5 }
▶ System.out.println("| bitti");
```

DEĞİŞKENLER

`i = (yok)`

ÇIKTI

```
1 2 4 5 7 | bitti
```

Etiketli `break dis`: iki döngüden birden çık

i = 1, j = 1: 1·1 = 1 ≠ 4, `1 1` yazılır.

Adım 1 / 6

```
1 dis:
2 for (int i = 1; i <= 3; i++) {
3     for (int j = 1; j <= 3; j++) {
4         if (i * j == 4) break dis;
▶         System.out.println(i + " " + j);
6     }
7 }
8 System.out.println("bitti");
```

DEĞİŞKENLER

`i = 1` `j = 1`

ÇIKTI

```
1 1
```

Etiketli `break dis`: iki döngüden birden çık

j = 2: 1·2 = 2 ≠ 4 → `1 2` yazılır.

Adım 2 / 6

```
1 dis:
2 for (int i = 1; i <= 3; i++) {
3     for (int j = 1; j <= 3; j++) {
4         if (i * j == 4) break dis;
▶         System.out.println(i + " " + j);
6     }
7 }
8 System.out.println("bitti");
```

DEĞİŞKENLER

`i = 1` `j = 2`

ÇIKTI

```
1 1
1 2
```

Etiketli `break dis`: iki döngüden birden çık

j = 3: 1·3 = 3 ≠ 4 → `1 3` yazılır; iç döngü biter.

Adım 3 / 6

```
1 dis:
2 for (int i = 1; i <= 3; i++) {
3     for (int j = 1; j <= 3; j++) {
4         if (i * j == 4) break dis;
▶         System.out.println(i + " " + j);
6     }
7 }
8 System.out.println("bitti");
```

DEĞİŞKENLER

`i = 1` `j = 3`

ÇIKTI

```
1 1
1 2
1 3
```

Etiketli `break dis`: iki döngüden birden çık

i = 2; iç döngü yeniden başlar: j = 1, 2·1 = 2 ≠ 4 → `2 1` yazılır.

Adım 4 / 6

```
1 dis:
2 for (int i = 1; i <= 3; i++) {
3     for (int j = 1; j <= 3; j++) {
4         if (i * j == 4) break dis;
▶         System.out.println(i + " " + j);
6     }
7 }
8 System.out.println("bitti");
```

DEĞİŞKENLER

`i = 2` `j = 1`

ÇIKTI

```
1 1
1 2
1 3
2 1
```

Etiketli `break dis`: iki döngüden birden çık

i = 2, j = 2: 2·2 = 4 → `break dis`. Yazdırmadan **her iki döngüden birden** çıkılır; i = 3 hiç denenmez.

Adım 5 / 6

```
1 dis:
2 for (int i = 1; i <= 3; i++) {
3     for (int j = 1; j <= 3; j++) {
          if (i * j == 4) break dis;
5         System.out.println(i + " " + j);
6     }
7 }
8 System.out.println("bitti");
```

DEĞİŞKENLER

`i = 2` `j = 2`

ÇIKTI

```
1 1
1 2
1 3
2 1
```

Etiketli `break dis`: iki döngüden birden çık

Etiketli break dış döngünün ardındaki satıra atlar: `bitti` yazılır. Çıktı: 4 sayı satırı + bitti.

Adım 6 / 6

```
1 dis:
2 for (int i = 1; i <= 3; i++) {
3     for (int j = 1; j <= 3; j++) {
4         if (i * j == 4) break dis;
5         System.out.println(i + " " + j);
6     }
7 }
▶ System.out.println("bitti");
```

DEĞİŞKENLER

`i = (yok)` `j = (yok)`

ÇIKTI

```
1 1
1 2
1 3
2 1
bitti
```

Karşılaştırma: düz `break` yalnızca iç döngüyü bitirir

i = 1: j = 1, 2, 3 için 1·j ≠ 4; `1 1` , `1 2` , `1 3` yazılır.

Adım 1 / 6

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
3         if (i * j == 4) break;
▶         System.out.println(i + " " + j);
5     }
6 }
7 System.out.println("bitti");
```

DEĞİŞKENLER

`i = 1` `j = 3`

ÇIKTI

```
1 1
1 2
1 3
```

Karşılaştırma: düz `break` yalnızca iç döngüyü bitirir

i = 2, j = 1: 2·1 = 2 ≠ 4 → `2 1` yazılır.

Adım 2 / 6

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
3         if (i * j == 4) break;
▶         System.out.println(i + " " + j);
5     }
6 }
7 System.out.println("bitti");
```

DEĞİŞKENLER

`i = 2` `j = 1`

ÇIKTI

```
1 1
1 2
1 3
2 1
```

Karşılaştırma: düz `break` yalnızca iç döngüyü bitirir

i = 2, j = 2: 2·2 = 4 → düz `break` yalnızca **iç** döngüyü bitirir; dış döngü devam eder.

Adım 3 / 6

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
▶           if (i * j == 4) break;
4           System.out.println(i + " " + j);
5       }
6   }
7   System.out.println("bitti");
```

DEĞİŞKENLER

`i = 2` `j = 2`

ÇIKTI

```
1 1
1 2
1 3
2 1
```

Karşılaştırma: düz `break` yalnızca iç döngüyü bitirir

i++ → 3; iç döngü yeniden başlar: j = 1 → `3 1` yazılır.

Adım 4 / 6

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
3         if (i * j == 4) break;
          System.out.println(i + " " + j);
5     }
6 }
7 System.out.println("bitti");
```

DEĞİŞKENLER

`i = 3` `j = 1`

ÇIKTI

```
1 1
1 2
1 3
2 1
3 1
```

Karşılaştırma: düz `break` yalnızca iç döngüyü bitirir

j = 2 ve j = 3: 3·2 = 6, 3·3 = 9 ≠ 4 → `3 2` ve `3 3` yazılır.

Adım 5 / 6

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
3         if (i * j == 4) break;
▶         System.out.println(i + " " + j);
5     }
6 }
7 System.out.println("bitti");
```

DEĞİŞKENLER

`i = 3` `j = 3`

ÇIKTI

```
1 1
1 2
1 3
2 1
3 1
3 2
3 3
```

Karşılaştırma: düz `break` yalnızca iç döngüyü bitirir

Dış döngü de biter, `bitti` yazılır. Düz break ile 7 sayı satırı, etiketli break ile 4 sayı satırı çıktı.

Adım 6 / 6

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
3         if (i * j == 4) break;
4         System.out.println(i + " " + j);
5     }
6 }
▶ System.out.println("bitti");
```

DEĞİŞKENLER

`i = (yok)` `j = (yok)`

ÇIKTI

```
1 1
1 2
1 3
2 1
3 1
3 2
3 3
bitti
```

Sonsuz Döngü




```
for (int i = 0; i < 5; i--) → i = 0, -1, -2, … koşul hep true

while (i < 5) { … } // i++ yok → i hep 0
```

- Koşul hiçbir zaman false olmuyorsa döngü bitmez: program “donmuş” görünür.
- Not: `i--` ile `int` sınırı aşılırsa değer sarar ve döngü yaklaşık 2,1 milyar turdan sonra biter; pratikte sonsuz sayılır.
- Sayaç yanlış yönde güncellenmiş ya da hiç güncellenmemiş olabilir.
- Kasıtlı `while (true)` döngülerinde de içeride bir `break` koşulu olmalıdır. Takılırsa programı Ctrl+C ile durdurun.

double Sayaç Tuzağı: `x += 0.1`

Başlangıç: **n = 0**. Amaç 0,1'er artıp 10 tur dönmek.

Adım 1 / 9

```
▶ int n = 0;
2 for (double x = 0; x < 1.0; x += 0.1) {
3     n++;
4 }
5 System.out.println(n);
```

DEĞİŞKENLER

n = 0

ÇIKTI

`x != 1.0` yazılsaydı x hiçbir zaman tam 1.0 olmayacağından döngü bitmezdi. Gerçek sayılarda sayaç yerine `int` kullanın.

double Sayaç Tuzağı: `x += 0.1`

`x = 0.0`, koşul `0.0 < 1.0` → **true**.

Adım 2 / 9

```
1 int n = 0;
▶ for (double x = 0; x < 1.0; x += 0.1) {
3     n++;
4 }
5 System.out.println(n);
```

DEĞİŞKENLER

`n = 0` `x = 0.0`

ÇIKTI

`x != 1.0` yazılsaydı x hiçbir zaman tam 1.0 olmayacağından döngü bitmezdi. Gerçek sayılarda sayaç yerine `int` kullanın.

double Sayaç Tuzağı: `x += 0.1`

Gövde: **n = 1**.

Adım 3 / 9

```
1 int n = 0;
2 for (double x = 0; x < 1.0; x += 0.1) {
▶     n++;
4 }
5 System.out.println(n);
```

DEĞİŞKENLER

n = 1 | x = 0.0

ÇIKTI

`x != 1.0` yazılsaydı x hiçbir zaman tam 1.0 olmayacağından döngü bitmezdi. Gerçek sayılarda sayaç yerine `int` kullanın.

double Sayaç Tuzağı: `x += 0.1`

Güncelleme: x = 0.1. 0,1 ikili tabanda tam gösterilemez; küçük hatalar birikir.

Adım 4 / 9

```
1 int n = 0;
▶ for (double x = 0; x < 1.0; x += 0.1) {
3     n++;
4 }
5 System.out.println(n);
```

DEĞİŞKENLER

`n = 1` `x = 0.1`

ÇIKTI

`x != 1.0` yazılsaydı x hiçbir zaman tam 1.0 olmayacağından döngü bitmezdi. Gerçek sayılarda sayaç yerine `int` kullanın.

double Sayaç Tuzağı: `x += 0.1`

10. turun sonu (arada 8 tur atlandı): **n = 10**, x = 0.8999999999999999.

Adım 5 / 9

```
1 int n = 0;
2 for (double x = 0; x < 1.0; x += 0.1) {
▶     n++;
4 }
5 System.out.println(n);
```

DEĞİŞKENLER

n = 10 | x = 0.8999999999999999

ÇIKTI

`x != 1.0` yazılsaydı x hiçbir zaman tam 1.0 olmayacağından döngü bitmezdi. Gerçek sayılarda sayaç yerine `int` kullanın.

double Sayaç Tuzağı: `x += 0.1`

Güncelleme: x = 0.9999999999999999, tam 1.0 **değil**. Koşul true: döngü bir tur fazla döner.

Adım 6 / 9

```
1 int n = 0;
▶ for (double x = 0; x < 1.0; x += 0.1) {
3     n++;
4 }
5 System.out.println(n);
```

DEĞİŞKENLER

n = 10 | x = 0.9999999999999999

ÇIKTI

`x != 1.0` yazılsaydı x hiçbir zaman tam 1.0 olmayacağından döngü bitmezdi. Gerçek sayılarda sayaç yerine `int` kullanın.

double Sayaç Tuzağı: `x += 0.1`

11. tur: **n = 11**.

Adım 7 / 9

```
1 int n = 0;
2 for (double x = 0; x < 1.0; x += 0.1) {
▶     n++;
4 }
5 System.out.println(n);
```

DEĞİŞKENLER

n = 11

x = 0.9999999999999999

ÇIKTI

`x != 1.0` yazılsaydı x hiçbir zaman tam 1.0 olmayacağından döngü bitmezdi. Gerçek sayılarda sayaç yerine `int` kullanın.

double Sayaç Tuzağı: `x += 0.1`

x = 1.0999999999999999, koşul false → döngü biter.

Adım 8 / 9

```
1 int n = 0;
▶ for (double x = 0; x < 1.0; x += 0.1) {
3     n++;
4 }
5 System.out.println(n);
```

DEĞİŞKENLER

n = 11 | x = 1.0999999999999999

ÇIKTI

`x != 1.0` yazılsaydı x hiçbir zaman tam 1.0 olmayacağından döngü bitmezdi. Gerçek sayılarda sayaç yerine `int` kullanın.

double Sayaç Tuzağı: `x += 0.1`

11 yazılır; beklenen 10 idi.

Adım 9 / 9

```
1 int n = 0;
2 for (double x = 0; x < 1.0; x += 0.1) {
3     n++;
4 }
▶ System.out.println(n);
```

DEĞİŞKENLER

`n = 11` `x = 1.0999999999999999`

ÇIKTI

```
11
```

`x != 1.0` yazılsaydı x hiçbir zaman tam 1.0 olmayacağından döngü bitmezdi. Gerçek sayılarda sayaç yerine `int` kullanın.

DOĞRULAMA

Döngü Değişmezi

- Her turun **başında** doğru kalan bir özelliktir.
- Toplam döngüsünde: `toplam = 1 + 2 + … + (i − 1)` .
- Döngü bitince (i = n + 1) değişmez, sonucun doğru olduğunu söyler.

turun başı: i = 1



Başlangıçta hiçbir sayı işlenmedi: toplam = 0 (boş toplam). Değişmez baştan doğru.

Adım 1 / 5

DOĞRULAMA

Döngü Değişmezi

- Her turun **başında** doğru kalan bir özelliktir.
- Toplam döngüsünde: `toplam = 1 + 2 + … + (i − 1)`.
- Döngü bitince (i = n + 1) değişmez, sonucun doğru olduğunu söyler.

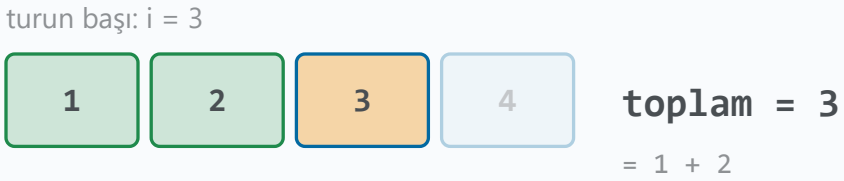


i = 2 turunun başında toplam, **1 … 1** sayılarının toplamı: 1.

DOĞRULAMA

Döngü Değişmezi

- Her turun **başında** doğru kalan bir özelliktir.
- Toplam döngüsünde: `toplam = 1 + 2 + … + (i − 1)`.
- Döngü bitince (i = n + 1) değişmez, sonucun doğru olduğunu söyler.



i = 3 turunun başında toplam, **1 … 2** sayılarının toplamı: 3.

DOĞRULAMA

Döngü Değişmezi

- Her turun **başında** doğru kalan bir özelliktir.
- Toplam döngüsünde: `toplam = 1 + 2 + … + (i − 1)` .
- Döngü bitince (i = n + 1) değişmez, sonucun doğru olduğunu söyler.

turun başı: i = 4



toplam = 6

= 1 + 2 + 3

i = 4 turunun başında toplam, **1 … 3** sayılarının toplamı: 6.

DOĞRULAMA

Döngü Değişmezi

- Her turun **başında** doğru kalan bir özelliktir.
- Toplam döngüsünde: `toplam = 1 + 2 + … + (i − 1)` .
- Döngü bitince (i = n + 1) değişmez, sonucun doğru olduğunu söyler.

turun başı: i = 5



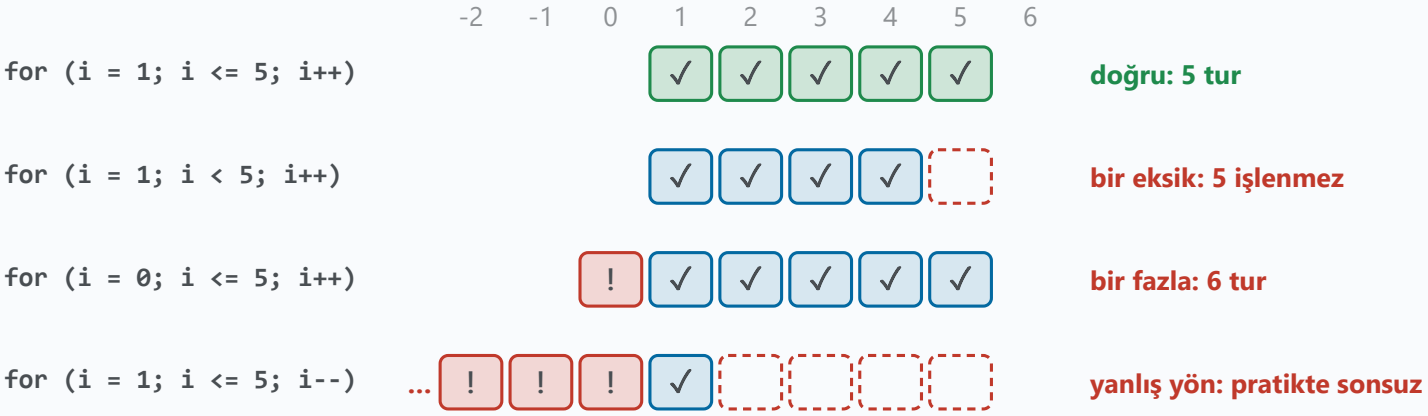
toplam = 10

= 1 + 2 + 3 + 4

i = 5 → koşul false. Değişmez hâlâ doğru: toplam = 1 + 2 + 3 + 4 = 10 → sonuç doğru.

DOĞRULAMA

Sınır Hataları (1’den 5’e say)



- `<` ile `<=`, 0 ile 1 başlangıcı arasındaki fark bir tur eksik/fazla demektir (“off-by-one”).
- Test: ilk turu, son turu ve tur sayısını elle kontrol edin.

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası Denetimi: `i < 5` → 5 tur

Başlangıç: **i = 0**, sayac = 0. Koşul `0 < 5` → true.

Adım 1 / 7

```
▶ int sayac = 0;
▶ for (int i = 0; i < 5; i++) {
3     System.out.print(i + " ");
4     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 0` `sayac = 0`

ÇIKTI

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası Denetimi: `i < 5` → 5 tur

Tur 1: i = 0 yazılır, sayac = 1 olur.

Adım 2 / 7

```
1 int sayac = 0;
2 for (int i = 0; i < 5; i++) {
      System.out.print(i + " ");
      sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 0` `sayac = 1`

ÇIKTI

```
0
```

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası Denetimi: `i < 5` → 5 tur

Tur 2: i = 1 yazılır, sayac = 2 olur.

Adım 3 / 7

```
1 int sayac = 0;
2 for (int i = 0; i < 5; i++) {
▶     System.out.print(i + " ");
▶     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 1` `sayac = 2`

ÇIKTI

```
0 1
```

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası Denetimi: `i < 5` → 5 tur

Tur 3: i = 2 yazılır, sayac = 3 olur.

Adım 4 / 7

```
1 int sayac = 0;
2 for (int i = 0; i < 5; i++) {
▶     System.out.print(i + " ");
▶     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 2` `sayac = 3`

ÇIKTI

```
0 1 2
```

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası Denetimi: `i < 5` → 5 tur

Tur 4: i = 3 yazılır, sayac = 4 olur.

Adım 5 / 7

```
1 int sayac = 0;
2 for (int i = 0; i < 5; i++) {
▶     System.out.print(i + " ");
▶     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 3` `sayac = 4`

ÇIKTI

```
0 1 2 3
```

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası Denetimi: `i < 5` → 5 tur

Tur 5: i = 4 yazılır, sayac = 5 olur.

Adım 6 / 7

```
1 int sayac = 0;
2 for (int i = 0; i < 5; i++) {
▶     System.out.print(i + " ");
▶     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 4` `sayac = 5`

ÇIKTI

```
0 1 2 3 4
```

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası Denetimi: `i < 5` → 5 tur

i++ → 5; `5 < 5` → false, döngü biter. Tam **5 tur**: i = 0…4.

Adım 7 / 7

```
1 int sayac = 0;
2 for (int i = 0; i < 5; i++) {
3     System.out.print(i + " ");
4     sayac++;
5 }
▶ System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = (yok)` `sayac = 5`

ÇIKTI

```
0 1 2 3 4 | tur: 5
```

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası: `i <= 5` → 6 tur

Başlangıç: **i = 0**, sayac = 0. Koşul `0 <= 5` → true.

Adım 1 / 8

```
▶ int sayac = 0;
▶ for (int i = 0; i <= 5; i++) {
3     System.out.print(i + " ");
4     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 0` `sayac = 0`

ÇIKTI

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası: `i <= 5` → 6 tur

Tur 1: i = 0 yazılır, sayac = 1 olur.

Adım 2 / 8

```
1 int sayac = 0;
2 for (int i = 0; i <= 5; i++) {
▶     System.out.print(i + " ");
▶     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 0` `sayac = 1`

ÇIKTI

```
0
```

Sınır Hatası: `i <= 5` → 6 tur

Tur 2: i = 1 yazılır, sayac = 2 olur.

Adım 3 / 8

```
1 int sayac = 0;
2 for (int i = 0; i <= 5; i++) {
▶     System.out.print(i + " ");
▶     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 1` `sayac = 2`

ÇIKTI

```
0 1
```

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası: `i <= 5` → 6 tur

Tur 3: i = 2 yazılır, sayac = 3 olur.

Adım 4 / 8

```
1 int sayac = 0;
2 for (int i = 0; i <= 5; i++) {
▶     System.out.print(i + " ");
▶     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 2` `sayac = 3`

ÇIKTI

```
0 1 2
```

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası: `i <= 5` → 6 tur

Tur 4: i = 3 yazılır, sayac = 4 olur.

Adım 5 / 8

```
1 int sayac = 0;
2 for (int i = 0; i <= 5; i++) {
▶     System.out.print(i + " ");
▶     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 3` `sayac = 4`

ÇIKTI

```
0 1 2 3
```

Sınır Hatası: `i <= 5` → 6 tur

Tur 5: i = 4 yazılır, sayac = 5 olur.

Adım 6 / 8

```
1 int sayac = 0;
2 for (int i = 0; i <= 5; i++) {
▶     System.out.print(i + " ");
▶     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 4` `sayac = 5`

ÇIKTI

```
0 1 2 3 4
```

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası: `i <= 5` → 6 tur

Tur 6: i = 5 yazılır, sayac = 6 olur.

Adım 7 / 8

```
1 int sayac = 0;
2 for (int i = 0; i <= 5; i++) {
▶     System.out.print(i + " ");
▶     sayac++;
5 }
6 System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = 5` `sayac = 6`

ÇIKTI

```
0 1 2 3 4 5
```

DOĞRULAMA · KODU ADIM ADIM ÇALIŞTIR

Sınır Hatası: `i <= 5` → 6 tur

i++ → 6; `6 <= 5` → false. `<=` yüzünden fazladan bir tur döndü: sayac = **6**, beklenen 5 değil.

Adım 8 / 8

```
1 int sayac = 0;
2 for (int i = 0; i <= 5; i++) {
3     System.out.print(i + " ");
4     sayac++;
5 }
▶ System.out.println("| tur: " + sayac);
```

DEĞİŞKENLER

`i = (yok)` `sayac = 6`

ÇIKTI

```
0 1 2 3 4 5 | tur: 6
```

ÖRNEK

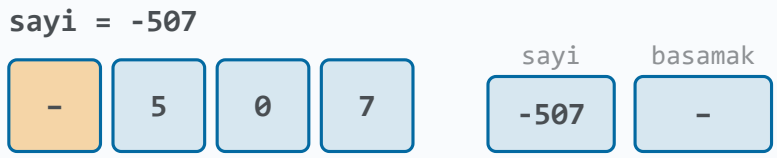
Basamak Sayısı

```
int sayi = -507;
long kalan = sayi;
if (kalan < 0) kalan = -kalan;
int basamak = 0;
do {
    basamak++;
    kalan /= 10;
} while (kalan > 0);
System.out.println(basamak);
```



- Her `/ 10` son basamağı atar; kaç kez bölündüğünü sayarız.
- **do-while** sayesinde 0 da 1 basamak sayılır; ayrı bir özel durum gerekmez.
- long kopya, `-2147483648` gibi uç değerde işaret değiştirirken taşmayı önler.

Kod İzleme: basamak sayısı (sayi = −507)



Negatif bir sayıyla deneyelim: **sayi = −507**.

Adım 1 / 14

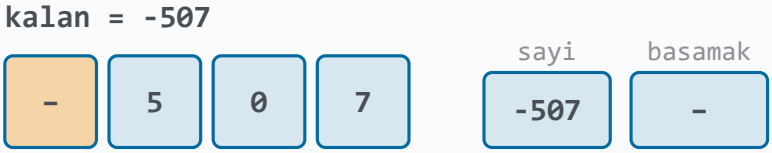
```
▶ int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
6     basamak++;
7     kalan /= 10;
8 } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = −507`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)



int değer long'a kopyalanır (−2147483648'in tersi int'e sığmaz, long'a sığar).

Adım 2 / 14

```
1 int sayi = -507;
▶ long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
6     basamak++;
7     kalan /= 10;
8 } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = -507` `kalan = -507`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)



−507 < 0 → true → **kalan = 507**. İşaret basamak sayısını değiştirmez.

Adım 3 / 14

```
1 int sayi = -507;
2 long kalan = sayi;
▶ if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
6     basamak++;
7     kalan /= 10;
8 } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = -507` `kalan = 507`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)



Sayaç **basamak = 0**.

Adım 4 / 14

```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
▶ int basamak = 0;
5 do {
6     basamak++;
7     kalan /= 10;
8 } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = −507` `kalan = 507` `basamak = 0`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)



Gövde: bir basamak sayıldı → basamak = **1**.

Adım 5 / 14

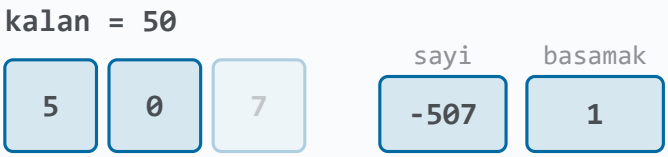
```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
▶     basamak++;
7     kalan /= 10;
8 } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = −507` `kalan = 507` `basamak = 1`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)



kalan = 507 / 10 = **50** (tam sayı bölmesi son basamağı atar).

Adım 6 / 14

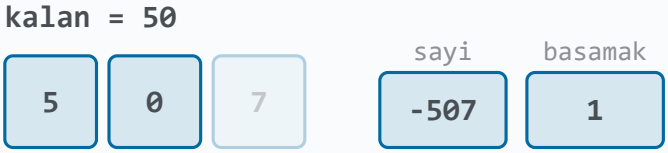
```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
6     basamak++;
▶     kalan /= 10;
8 } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = -507` `kalan = 50` `basamak = 1`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)



Koşul: 50 > 0 → **true** → tekrar.

Adım 7 / 14

```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
6     basamak++;
7     kalan /= 10;
▶ } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = -507` `kalan = 50` `basamak = 1`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)



Gövde: bir basamak sayıldı → basamak = **2**.

Adım 8 / 14

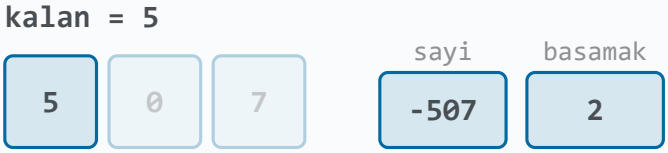
```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
▶     basamak++;
7     kalan /= 10;
8 } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = −507` `kalan = 50` `basamak = 2`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)



kalan = 50 / 10 = **5** (tam sayı bölmesi son basamağı atar).

Adım 9 / 14

```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
6     basamak++;
▶     kalan /= 10;
8 } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = -507` `kalan = 5` `basamak = 2`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)

kalan = 5

5	0	7

sayi	basamak
-507	2

Koşul: 5 > 0 → **true** → tekrar.

Adım 10 / 14

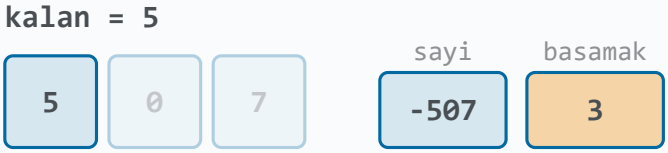
```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
6     basamak++;
7     kalan /= 10;
▶ } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = -507` `kalan = 5` `basamak = 2`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)



Gövde: bir basamak sayıldı → basamak = **3**.

Adım 11 / 14

```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
▶     basamak++;
7     kalan /= 10;
8 } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = −507` `kalan = 5` `basamak = 3`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)

kalan = 0

5	0	7

sayi	basamak
-507	3

kalan = 5 / 10 = **0** (tam sayı bölmesi son basamağı atar).

Adım 12 / 14

```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
6     basamak++;
▶     kalan /= 10;
8 } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = -507` `kalan = 0` `basamak = 3`

ÇIKTI

Kod İzleme: basamak sayısı (sayi = −507)

kalan = 0

5	0	7

sayi	basamak
-507	3

Koşul: 0 > 0 → **false** → bitti.

Adım 13 / 14

```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
6     basamak++;
7     kalan /= 10;
▶ } while (kalan > 0);
9 System.out.println(basamak);
```

DEĞİŞKENLER

`sayi = -507` `kalan = 0` `basamak = 3`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: basamak sayısı (sayi = −507)

kalan = 0

5	0	7

sayi	basamak
-507	3

3 yazılır. sayi = 0 olsaydı do-while gövdeyi yine bir kez çalıştırır → 1 basamak (doğru).

Adım 14 / 14

```
1 int sayi = -507;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int basamak = 0;
5 do {
6     basamak++;
7     kalan /= 10;
8 } while (kalan > 0);
▶ System.out.println(basamak);
```

DEĞİŞKENLER

sayi = **−507** | kalan = **0** | basamak = **3**

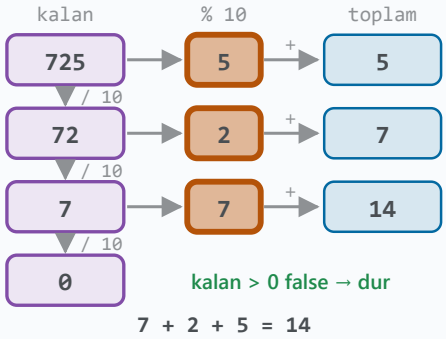
ÇIKTI

```
3
```

ÖRNEK

Rakamlar Toplamı

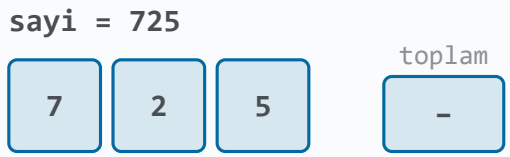
```
int sayi = 725;
long kalan = sayi;
if (kalan < 0) kalan = -kalan;
int toplam = 0;
while (kalan > 0) {
    toplam += (int) (kalan % 10);
    kalan /= 10;
}
System.out.println(toplam);
```



- `% 10` son rakamı verir.
- `/ 10` işlenen son rakamı sayıdan çıkarır.
- Burada while yeterli: 0 için gövde hiç çalışmaz, toplam 0 kalır (doğru).

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: rakamlar toplamı (sayi = 725)



Rakamları 7, 2 ve 5 olan sayı: **725**.

Adım 1 / 15

```
▶ int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
5 while (kalan > 0) {
6     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

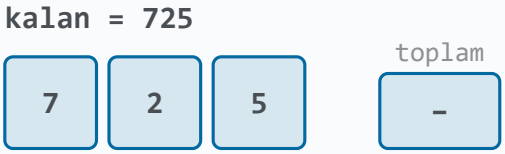
DEĞİŞKENLER

`sayi = 725`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: rakamlar toplamı (sayi = 725)



Üzerinde çalışmak için kopya: kalan = 725.

Adım 2 / 15

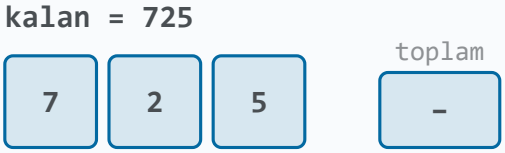
```
1 int sayi = 725;
▶ long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
5 while (kalan > 0) {
6     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

DEĞİŞKENLER

sayi = 725 kalan = 725

ÇIKTI

Kod İzleme: rakamlar toplamı (sayi = 725)



725 < 0 → false: işaret değişmez.

Adım 3 / 15

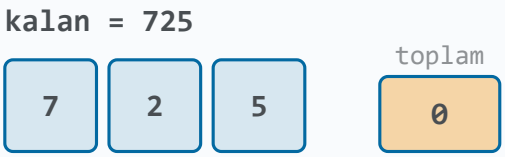
```
1 int sayi = 725;
2 long kalan = sayi;
▶ if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
5 while (kalan > 0) {
6     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

DEĞİŞKENLER

`sayi = 725` `kalan = 725`

ÇIKTI

Kod İzleme: rakamlar toplamı (sayi = 725)



Biriktirici **toplam = 0**.

Adım 4 / 15

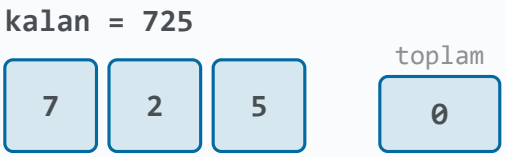
```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
▶ int toplam = 0;
5 while (kalan > 0) {
6     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

DEĞİŞKENLER

`sayi = 725` `kalan = 725` `toplam = 0`

ÇIKTI

Kod İzleme: rakamlar toplamı (sayi = 725)



Koşul: 725 > 0 → **true**.

Adım 5 / 15

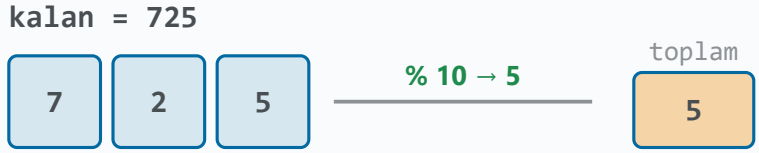
```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
▶ while (kalan > 0) {
6     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

DEĞİŞKENLER

`sayi = 725` `kalan = 725` `toplam = 0`

ÇIKTI

Kod İzleme: rakamlar toplamı (sayi = 725)



Son rakam 725 % 10 = **5**; toplam = 0 + 5 = **5**.

Adım 6 / 15

```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
5 while (kalan > 0) {
▶     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

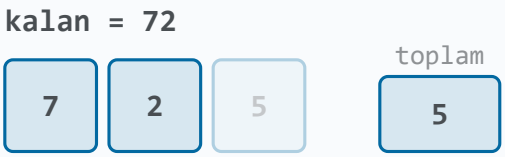
DEĞİŞKENLER

`sayi = 725` `kalan = 725` `toplam = 5`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: rakamlar toplamı (sayi = 725)



kalan = 725 / 10 = **72**: işlenen rakam atıldı.

Adım 7 / 15

```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
5 while (kalan > 0) {
6     toplam += (int) (kalan % 10);
▶     kalan /= 10;
8 }
9 System.out.println(toplam);
```

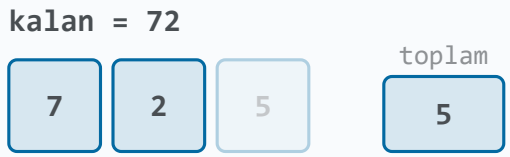
DEĞİŞKENLER

sayi = **725** kalan = **72** toplam = **5**

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: rakamlar toplamı (sayi = 725)



Koşul: 72 > 0 → **true**.

Adım 8 / 15

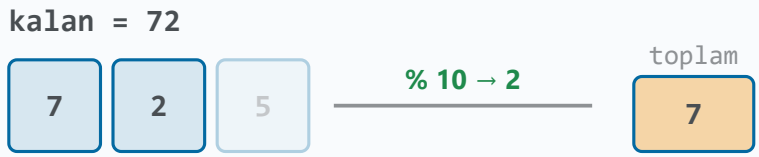
```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
▶ while (kalan > 0) {
6     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

DEĞİŞKENLER

sayi = **725** | kalan = **72** | toplam = **5**

ÇIKTI

Kod İzleme: rakamlar toplamı (sayi = 725)



Son rakam 72 % 10 = **2**; toplam = 5 + 2 = **7**.

Adım 9 / 15

```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
5 while (kalan > 0) {
▶     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

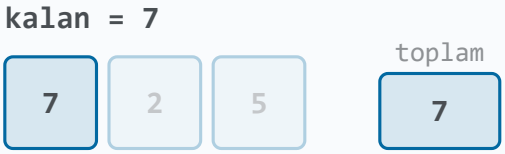
DEĞİŞKENLER

`sayi = 725` `kalan = 72` `toplam = 7`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: rakamlar toplamı (sayi = 725)



kalan = 72 / 10 = **7**: işlenen rakam atıldı.

Adım 10 / 15

```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
5 while (kalan > 0) {
6     toplam += (int) (kalan % 10);
▶     kalan /= 10;
8 }
9 System.out.println(toplam);
```

DEĞİŞKENLER

sayi = **725** kalan = **7** toplam = **7**

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: rakamlar toplamı (sayi = 725)



Koşul: 7 > 0 → **true**.

Adım 11 / 15

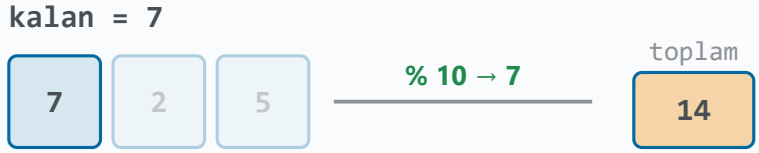
```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
▶ while (kalan > 0) {
6     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

DEĞİŞKENLER

`sayi = 725` `kalan = 7` `toplam = 7`

ÇIKTI

Kod İzleme: rakamlar toplamı (sayi = 725)



Son rakam 7 % 10 = **7**; toplam = 7 + 7 = **14**.

Adım 12 / 15

```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
5 while (kalan > 0) {
▶     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

DEĞİŞKENLER

`sayi = 725` `kalan = 7` `toplam = 14`

ÇIKTI

Kod İzleme: rakamlar toplamı (sayi = 725)



kalan = 7 / 10 = **0**: işlenen rakam atıldı.

Adım 13 / 15

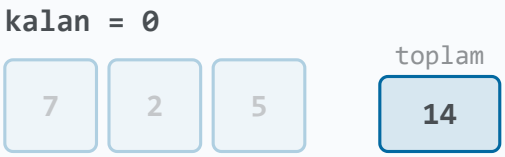
```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
5 while (kalan > 0) {
6     toplam += (int) (kalan % 10);
▶     kalan /= 10;
8 }
9 System.out.println(toplam);
```

DEĞİŞKENLER

`sayi = 725` `kalan = 0` `toplam = 14`

ÇIKTI

Kod İzleme: rakamlar toplamı (sayi = 725)



Koşul: 0 > 0 → **false**.

Adım 14 / 15

```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
▶ while (kalan > 0) {
6     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
9 System.out.println(toplam);
```

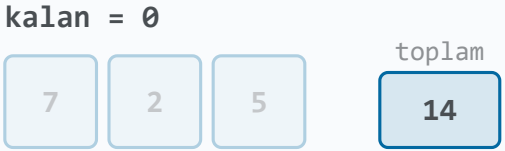
DEĞİŞKENLER

`sayi = 725` `kalan = 0` `toplam = 14`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: rakamlar toplamı (sayi = 725)



14 yazılır: 7 + 2 + 5 = 14. `sayi` değişmedi, yalnızca kopyası (`kalan`) tüketildi.

Adım 15 / 15

```
1 int sayi = 725;
2 long kalan = sayi;
3 if (kalan < 0) kalan = -kalan;
4 int toplam = 0;
5 while (kalan > 0) {
6     toplam += (int) (kalan % 10);
7     kalan /= 10;
8 }
▶ System.out.println(toplam);
```

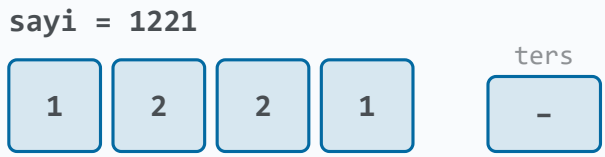
DEĞİŞKENLER

sayi = **725** kalan = **0** toplam = **14**

ÇIKTI

```
14
```

Sayıyı Ters Çevirme ve Palindrom (1221)



Rakamları tersten okununca da aynı kalan sayıya **palindrom** denir. Deneyelim: 1221.

Adım 1 / 17

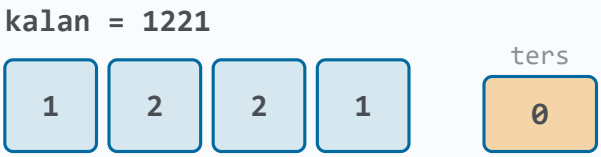
```
▶ int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



Kopya üzerinde çalışılır; ters sayı 0'dan kurulacak.

Adım 2 / 17

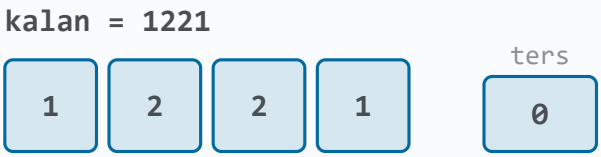
```
1 int sayi = 1221;
▶ int kalan = sayi, ters = 0;
3 while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 1221` `ters = 0`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



Koşul: 1221 > 0 → **true**.

Adım 3 / 17

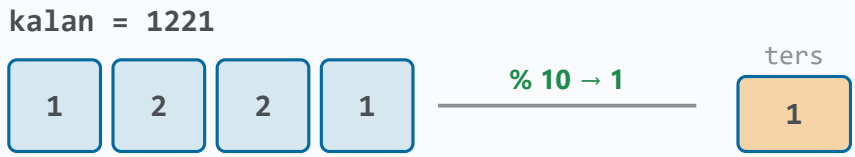
```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
▶ while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 1221` `ters = 0`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



rakam = 1221 % 10 = 1; ters = 0 × 10 + 1 = **1** (eski rakamlar bir basamak sola kayar).

Adım 4 / 17

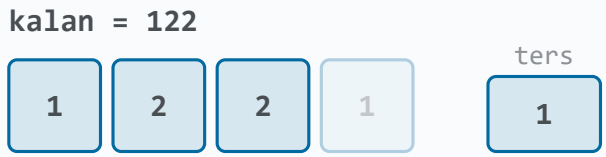
```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
▶     int rakam = kalan % 10;
▶     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

sayi = **1221** | kalan = **1221** | ters = **1** | rakam = **1**

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



kalan = 1221 / 10 = **122**.

Adım 5 / 17

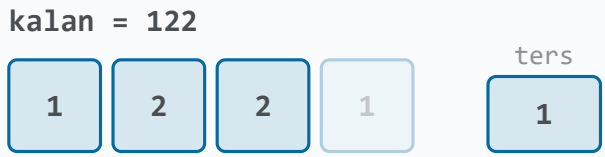
```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
▶     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 122` `ters = 1` `rakam = 1`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



Koşul: 122 > 0 → **true**.

Adım 6 / 17

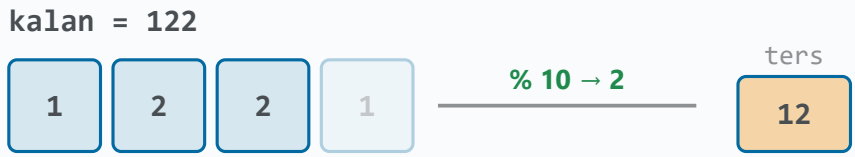
```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
▶ while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 122` `ters = 1`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



rakam = 122 % 10 = 2; ters = 1 × 10 + 2 = **12** (eski rakamlar bir basamak sola kayar).

Adım 7 / 17

```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
▶     int rakam = kalan % 10;
▶     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 122` `ters = 12` `rakam = 2`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



kalan = 122 / 10 = **12**.

Adım 8 / 17

```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
▶     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 12` `ters = 12` `rakam = 2`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



Koşul: 12 > 0 → **true**.

Adım 9 / 17

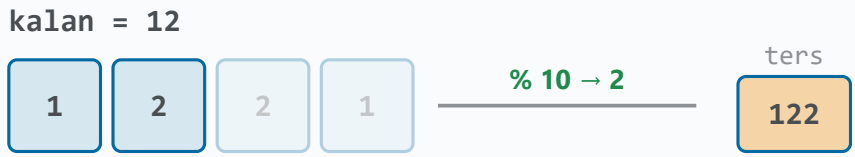
```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
▶ while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 12` `ters = 12`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



rakam = 12 % 10 = 2; ters = 12 × 10 + 2 = **122** (eski rakamlar bir basamak sola kayar).

Adım 10 / 17

```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
▶     int rakam = kalan % 10;
▶     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 12` `ters = 122` `rakam = 2`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



kalan = 12 / 10 = **1**.

Adım 11 / 17

```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
▶     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 1` `ters = 122` `rakam = 2`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



Koşul: 1 > 0 → **true**.

Adım 12 / 17

```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
▶ while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 1` `ters = 122`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



rakam = 1 % 10 = 1; ters = 122 × 10 + 1 = **1221** (eski rakamlar bir basamak sola kayar).

Adım 13 / 17

```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
▶     int rakam = kalan % 10;
▶     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

sayi = **1221** | kalan = **1** | ters = **1221** | rakam = **1**

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



kalan = 1 / 10 = **0**.

Adım 14 / 17

```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
▶     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

sayi = **1221** | kalan = **0** | ters = **1221** | rakam = **1**

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



Koşul: 0 > 0 → **false**.

Adım 15 / 17

```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
▶ while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 0` `ters = 1221`

ÇIKTI

Sayıyı Ters Çevirme ve Palindrom (1221)



Ters sayı **1221** yazılır.

Adım 16 / 17

```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
▶ System.out.println(ters);
9 System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 0` `ters = 1221`

ÇIKTI

```
1221
```

Sayıyı Ters Çevirme ve Palindrom (1221)



1221 == 1221 → **true**: 1221 bir palindromdur. (123 için ters 321 olur → false.)

Adım 17 / 17

```
1 int sayi = 1221;
2 int kalan = sayi, ters = 0;
3 while (kalan > 0) {
4     int rakam = kalan % 10;
5     ters = ters * 10 + rakam;
6     kalan /= 10;
7 }
8 System.out.println(ters);
▶ System.out.println(ters == sayi);  // palindrom mu?
```

DEĞİŞKENLER

`sayi = 1221` `kalan = 0` `ters = 1221`

ÇIKTI

```
1221
true
```

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3



Armstrong sayısı: rakamlarının (basamak sayısı kadar) kuvvetleri toplamı kendisine eşit. 153 üç basamaklı → küpler.

Adım 1 / 13

```
▶ int sayi = 153;
2 int kalan = sayi, toplam = 0;
3 while (kalan > 0) {
4     int r = kalan % 10;
5     toplam += r * r * r;   // 3 basamak: küp
6     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

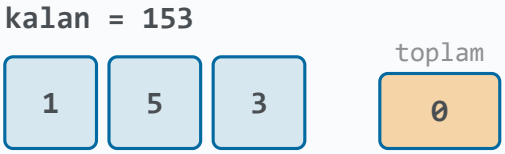
DEĞİŞKENLER

sayi = **153**

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3



kalan = 153, toplam = 0.

Adım 2 / 13

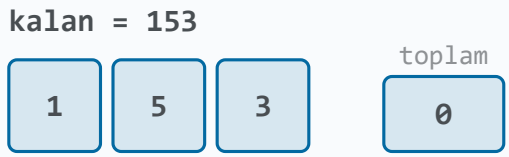
```
1 int sayi = 153;
▶ int kalan = sayi, toplam = 0;
3 while (kalan > 0) {
4     int r = kalan % 10;
5     toplam += r * r * r;   // 3 basamak: küp
6     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

sayi = 153 | kalan = 153 | toplam = 0

ÇIKTI

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3



Koşul: 153 > 0 → **true**.

Adım 3 / 13

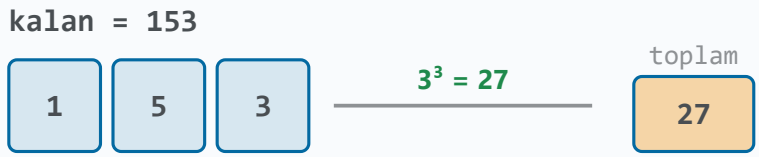
```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
▶ while (kalan > 0) {
4     int r = kalan % 10;
5     toplam += r * r * r;   // 3 basamak: küp
6     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

`sayi = 153` `kalan = 153` `toplam = 0`

ÇIKTI

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3



r = 3; toplam = 0 + 3×3×3 = **27**.

Adım 4 / 13

```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
3 while (kalan > 0) {
▶     int r = kalan % 10;
▶     toplam += r * r * r;   // 3 basamak: küp
6     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

sayi = **153** | kalan = **153** | toplam = **27** | r = **3**

ÇIKTI

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3



kalan /= 10 → 15.

Adım 5 / 13

```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
3 while (kalan > 0) {
4     int r = kalan % 10;
5     toplam += r * r * r;   // 3 basamak: küp
▶     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

sayi = **153** | kalan = **15** | toplam = **27** | r = **3**

ÇIKTI

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3



Koşul: 15 > 0 → **true**.

Adım 6 / 13

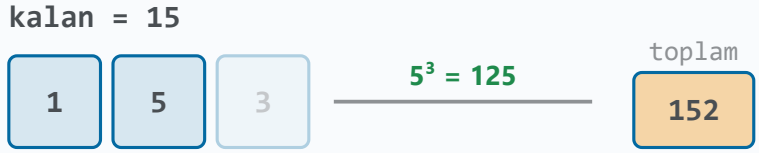
```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
▶ while (kalan > 0) {
4     int r = kalan % 10;
5     toplam += r * r * r;   // 3 basamak: küp
6     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

`sayi = 153` `kalan = 15` `toplam = 27`

ÇIKTI

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3



r = 5; toplam = 27 + 5×5×5 = **152**.

Adım 7 / 13

```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
3 while (kalan > 0) {
▶     int r = kalan % 10;
▶     toplam += r * r * r;   // 3 basamak: küp
6     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

`sayi = 153` `kalan = 15` `toplam = 152` `r = 5`

ÇIKTI

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3

kalan = 1

1	5	3

toplam

152

kalan /= 10 → 1.


```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
3 while (kalan > 0) {
4     int r = kalan % 10;
5     toplam += r * r * r;   // 3 basamak: küp
▶     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

sayi = 153 | kalan = 1 | toplam = 152 | r = 5

ÇIKTI

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3

kalan = 1

1	5	3

toplam

152

Koşul: 1 > 0 → **true**.

Adım 9 / 13

```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
▶ while (kalan > 0) {
4     int r = kalan % 10;
5     toplam += r * r * r;   // 3 basamak: küp
6     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

`sayi = 153` `kalan = 1` `toplam = 152`

ÇIKTI

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3



r = 1; toplam = 152 + 1×1×1 = **153**.

Adım 10 / 13

```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
3 while (kalan > 0) {
▶     int r = kalan % 10;
▶     toplam += r * r * r;   // 3 basamak: küp
6     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

`sayi = 153` `kalan = 1` `toplam = 153` `r = 1`

ÇIKTI

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3

kalan = 0

1	5	3

toplam

153

kalan /= 10 → 0.

Adım 11 / 13

```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
3 while (kalan > 0) {
4     int r = kalan % 10;
5     toplam += r * r * r;   // 3 basamak: küp
▶     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

`sayi = 153` `kalan = 0` `toplam = 153` `r = 1`

ÇIKTI

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3

kalan = 0

1	5	3

toplam

153

Koşul: 0 > 0 → **false**.

Adım 12 / 13

```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
▶ while (kalan > 0) {
4     int r = kalan % 10;
5     toplam += r * r * r;   // 3 basamak: küp
6     kalan /= 10;
7 }
8 System.out.println(toplam == sayi);
```

DEĞİŞKENLER

`sayi = 153` `kalan = 0` `toplam = 153`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Armstrong Sayısı: 153 = 1^3 + 5^3 + 3^3

kalan = 0

1	5	3

toplam

153

1 + 125 + 27 = 153 → **true**. Diğer 3 basamaklı Armstrong sayıları: 370, 371, 407.

Adım 13 / 13

```
1 int sayi = 153;
2 int kalan = sayi, toplam = 0;
3 while (kalan > 0) {
4     int r = kalan % 10;
5     toplam += r * r * r;   // 3 basamak: küp
6     kalan /= 10;
7 }
▶ System.out.println(toplam == sayi);
```

DEĞİŞKENLER

sayi = 153 | kalan = 0 | toplam = 153

ÇIKTI

```
true
```

Mükemmel Sayı: 28

`toplam = 0`

1	2	3	4	5	6	7	8	9	10	11	12	13	14

Mükemmel sayı: kendisi hariç bölenlerinin toplamı kendisine eşit (6 = 1 + 2 + 3). 28'i deneyelim.

Adım 1 / 18

```
▶ int n = 28;
2 int toplam = 0;
3 for (int i = 1; i <= n / 2; i++) {
4     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

n = **28**

ÇIKTI

Mükemmel Sayı: 28

`toplam = 0`

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Bölenler toplamı 0'dan başlar. n'nin kendisi hariç en büyük böleni en fazla n / 2 = 14.

Adım 2 / 18

```
1 int n = 28;
▶ int toplam = 0;
3 for (int i = 1; i <= n / 2; i++) {
4     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 0`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1



i = 1: 28 % 1 = 0 → bölen, toplam = **1**.

Adım 3 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 1` `i = 1`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 = 3



i = 2: 28 % 2 = 0 → bölen, toplam = **3**.

Adım 4 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 3` `i = 2`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 = 3



i = 3: 28 % 3 = 1 → bölmüyor.

Adım 5 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 3` `i = 3`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 = 7



i = 4: 28 % 4 = 0 → bölen, toplam = **7**.

Adım 6 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

n = 28 | toplam = 7 | i = 4

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 = 7



i = 5: 28 % 5 = 3 → bölmüyor.

Adım 7 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 7` `i = 5`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 = 7



i = 6: 28 % 6 = 4 → bölmüyor.

Adım 8 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

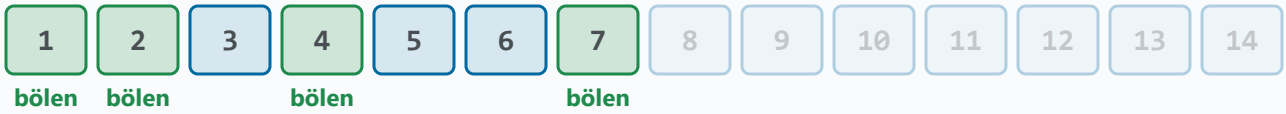
DEĞİŞKENLER

`n = 28` `toplam = 7` `i = 6`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 + 7 = 14



i = 7: 28 % 7 = 0 → bölen, toplam = **14**.

Adım 9 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

n = **28** toplam = **14** i = **7**

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 + 7 = 14



i = 8: 28 % 8 = 4 → bölmüyor.

Adım 10 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 14` `i = 8`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 + 7 = 14



i = 9: 28 % 9 = 1 → bölmüyor.

Adım 11 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 14` `i = 9`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 + 7 = 14



i = 10: 28 % 10 = 8 → bölmüyor.

Adım 12 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 14` `i = 10`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 + 7 = 14



i = 11: 28 % 11 = 6 → bölmüyor.

Adım 13 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 14` `i = 11`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 + 7 = 14



i = 12: 28 % 12 = 4 → bölmüyor.

Adım 14 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 14` `i = 12`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 + 7 = 14



i = 13: 28 % 13 = 2 → bölmüyor.

Adım 15 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 14` `i = 13`

ÇIKTI

Mükemmel Sayı: 28

toplam = 1 + 2 + 4 + 7 + 14 = 28



i = 14: 28 % 14 = 0 → bölen, toplam = **28**.

Adım 16 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
▶     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 28` `i = 14`

ÇIKTI



Mükemmel Sayı: 28

toplam = 1 + 2 + 4 + 7 + 14 = 28



i = 15 <= 14 → false: döngü biter.

Adım 17 / 18

```
1 int n = 28;
2 int toplam = 0;
▶ for (int i = 1; i <= n / 2; i++) {
4     if (n % i == 0) toplam += i;
5 }
6 System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 28`

ÇIKTI

```
true
```

Mükemmel Sayı: 28

`toplam = 1 + 2 + 4 + 7 + 14 = 28`



1 + 2 + 4 + 7 + 14 = 28 → **true**. Sonraki mükemmel sayılar: 496, 8128.

Adım 18 / 18

```
1 int n = 28;
2 int toplam = 0;
3 for (int i = 1; i <= n / 2; i++) {
4     if (n % i == 0) toplam += i;
5 }
▶ System.out.println(toplam == n);
```

DEĞİŞKENLER

`n = 28` `toplam = 28`

ÇIKTI

```
true
```

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Onluktan İkiliye Çevirme: 13 → 1101

bölme

kalan (bit)

bitler aşağıdan yukarı okunur

ikili: –

basamak: –

13'ü ikiliye çevireceğiz: sürekli 2'ye bölüp kalanları topluyoruz.

Adım 1 / 16

```
▶ int n = 13;
2 int ikili = 0, basamak = 1;
3 while (n > 0) {
4     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);   // 1101
```

DEĞİŞKENLER

n = 13

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Onluktan İkiliye Çevirme: 13 → 1101

bölme

kalan (bit)

bitler aşağıdan yukarı okunur

ikili	basamak
0	1

ikili sonucu ondalık rakamlarla (1 ve 0) kuracağız; basamak = 1 birler basamağı.

Adım 2 / 16

```
1 int n = 13;
▶ int ikili = 0, basamak = 1;
3 while (n > 0) {
4     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

n = 13 | ikili = 0 | basamak = 1

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme

kalan (bit)

bitler aşağıdan yukarı okunur

ikili	basamak
0	1

Koşul: 13 > 0 → **true**.

Adım 3 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
▶ while (n > 0) {
4     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 13` `ikili = 0` `basamak = 1`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme

13 / 2 = 6

kalan (bit)

1 ← bit

bitler aşağıdan yukarı okunur

ikili	basamak
1	1

13 % 2 = **1** → ikili += 1 × 1 → **1**.

Adım 4 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
3 while (n > 0) {
▶     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 13` `ikili = 1` `basamak = 1`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme

13 / 2 = 6

kalan (bit)

1

bitler aşağıdan yukarı okunur

ikili	basamak
1	10

Sonraki bit bir basamak solda: basamak = 10; n = n / 2 = **6**.

Adım 5 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
3 while (n > 0) {
4     ikili += (n % 2) * basamak;
      basamak *= 10;
      n /= 2;
7 }
8 System.out.println(ikili);   // 1101
```

DEĞİŞKENLER

`n = 6` `ikili = 1` `basamak = 10`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme

`13 / 2 = 6`

kalan (bit)

`1`

bitler aşağıdan yukarı okunur

ikili	basamak
1	10

Koşul: 6 > 0 → **true**.

Adım 6 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
▶ while (n > 0) {
4     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 6` `ikili = 1` `basamak = 10`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme

13 / 2 = 6

6 / 2 = 3

kalan (bit)

1

0 ← bit

bitler aşağıdan yukarı okunur

ikili	basamak
1	10

6 % 2 = **0** → ikili += 0 × 10 → **1**.

Adım 7 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
3 while (n > 0) {
▶     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 6` `ikili = 1` `basamak = 10`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme

13 / 2 = 6

6 / 2 = 3

kalan (bit)

1

0

bitler aşağıdan yukarı okunur

ikili	basamak
1	100

Sonraki bit bir basamak solda: basamak = 100; n = n / 2 = **3**.

Adım 8 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
3 while (n > 0) {
4     ikili += (n % 2) * basamak;
      basamak *= 10;
      n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 3` `ikili = 1` `basamak = 100`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Onluktan İkiliye Çevirme: 13 → 1101

bölme	kalan (bit)
13 / 2 = 6	1
6 / 2 = 3	0

bitler aşağıdan yukarı okunur

ikili	basamak
1	100

Koşul: 3 > 0 → **true**.

Adım 9 / 16

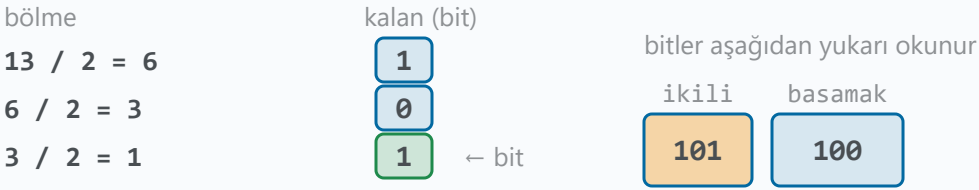
```
1 int n = 13;
2 int ikili = 0, basamak = 1;
▶ while (n > 0) {
4     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 3` `ikili = 1` `basamak = 100`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101



3 % 2 = **1** → ikili += 1 × 100 → **101**.

Adım 10 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
3 while (n > 0) {
▶     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 3` `ikili = 101` `basamak = 100`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme	kalan (bit)
13 / 2 = 6	1
6 / 2 = 3	0
3 / 2 = 1	1

bitler aşağıdan yukarı okunur

ikili	basamak
101	1000

Sonraki bit bir basamak solda: basamak = 1000; n = n / 2 = **1**.

Adım 11 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
3 while (n > 0) {
4     ikili += (n % 2) * basamak;
      basamak *= 10;
      n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 1` `ikili = 101` `basamak = 1000`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme	kalan (bit)
13 / 2 = 6	1
6 / 2 = 3	0
3 / 2 = 1	1

bitler aşağıdan yukarı okunur

ikili	basamak
101	1000

Koşul: 1 > 0 → **true**.

Adım 12 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
▶ while (n > 0) {
4     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 1` `ikili = 101` `basamak = 1000`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme	kalan (bit)
13 / 2 = 6	1
6 / 2 = 3	0
3 / 2 = 1	1
1 / 2 = 0	1 ← bit

bitler aşağıdan yukarı okunur

ikili	basamak
1101	1000

1 % 2 = **1** → ikili += 1 × 1000 → **1101**.

Adım 13 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
3 while (n > 0) {
▶     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 1` `ikili = 1101` `basamak = 1000`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme	kalan (bit)
13 / 2 = 6	1
6 / 2 = 3	0
3 / 2 = 1	1
1 / 2 = 0	1

bitler aşağıdan yukarı okunur

ikili	basamak
1101	10000

Sonraki bit bir basamak solda: basamak = 10000; n = n / 2 = **0**.

Adım 14 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
3 while (n > 0) {
4     ikili += (n % 2) * basamak;
      basamak *= 10;
      n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 0` `ikili = 1101` `basamak = 10000`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme	kalan (bit)
13 / 2 = 6	1
6 / 2 = 3	0
3 / 2 = 1	1
1 / 2 = 0	1

bitler aşağıdan yukarı okunur

ikili	basamak
1101	10000

Koşul: 0 > 0 → **false**.

Adım 15 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
▶ while (n > 0) {
4     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
8 System.out.println(ikili);    // 1101
```

DEĞİŞKENLER

`n = 0` `ikili = 1101` `basamak = 10000`

ÇIKTI

Onluktan İkiliye Çevirme: 13 → 1101

bölme	kalan (bit)
13 / 2 = 6	1
6 / 2 = 3	0
3 / 2 = 1	1
1 / 2 = 0	1

bitler aşağıdan yukarı okunur

ikili	basamak
1101	10000

1101 yazılır: 1·8 + 1·4 + 0·2 + 1·1 = 13. (Büyük sayılarda int taşar; sonuç String ile kurulabilir.)

Adım 16 / 16

```
1 int n = 13;
2 int ikili = 0, basamak = 1;
3 while (n > 0) {
4     ikili += (n % 2) * basamak;
5     basamak *= 10;
6     n /= 2;
7 }
▶ System.out.println(ikili);   // 1101
```

DEĞİŞKENLER

`n = 0` `ikili = 1101` `basamak = 10000`

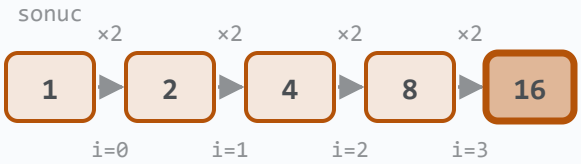
ÇIKTI

```
1101
```

ÖRNEK

Üs Alma

```
int taban = 2, us = 4;
int sonuc = 1;
for (int i = 0; i < us; i++) {
    sonuc *= taban;
}
System.out.println(sonuc);
```

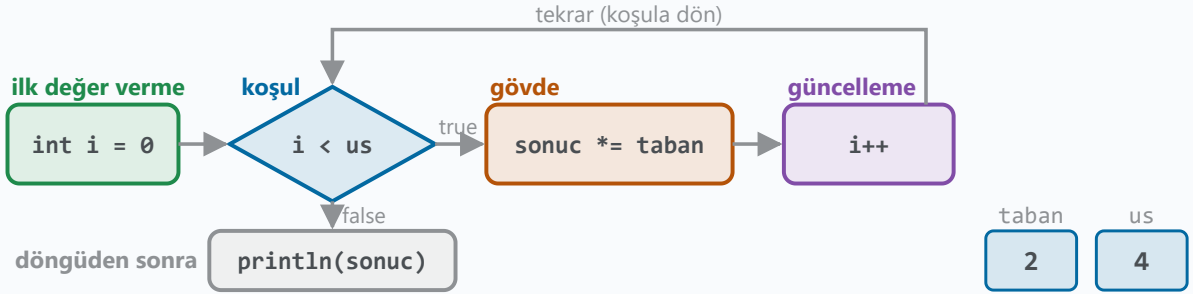


i = 4: 4 < 4 false → dur

2 × 2 × 2 × 2 = 16

- Döngü tam **üs** kez çalışır; her tur bir kez tabanla çarpar.
- us = 0 ise döngü hiç çalışmaz, sonuç 1 kalır ($2^0 = 1$).
- Negatif üsleri kapsamaz; büyük sonuçlar int sınırını aşar.

Kod İzleme: 2^4



2^4 hesaplanacak: taban = 2, us = 4.

Adım 1 / 17

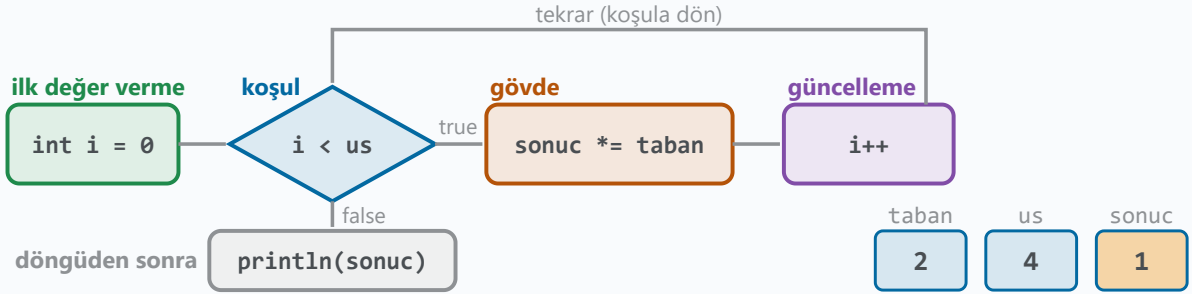
```
▶ int taban = 2, us = 4;
2 int sonuc = 1;
3 for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

taban = 2 | us = 4

ÇIKTI

Kod İzleme: 2^4



Çarpımın etkisiz elemanı **1**: sonuc = 1 (= 2^0).

Adım 2 / 17

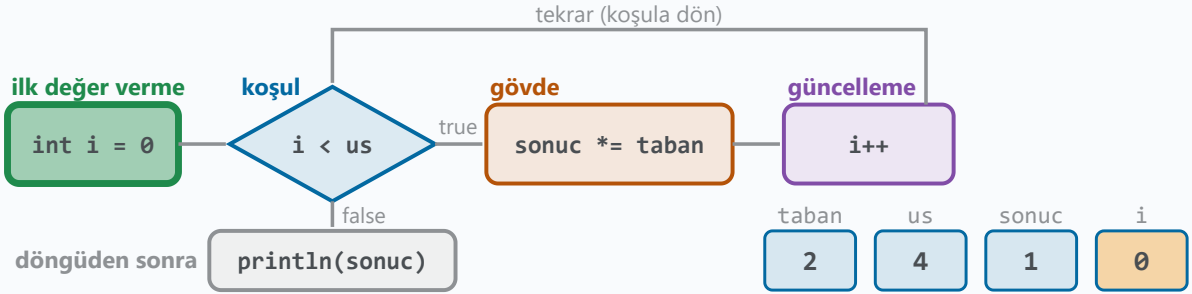
```
1 int taban = 2, us = 4;
▶ int sonuc = 1;
3 for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

taban = 2 | us = 4 | sonuc = 1

ÇIKTI

Kod İzleme: 2^4



İlk değer verme: i = 0. Döngü üs kez (0, 1, 2, 3) dönecek.

Adım 3 / 17

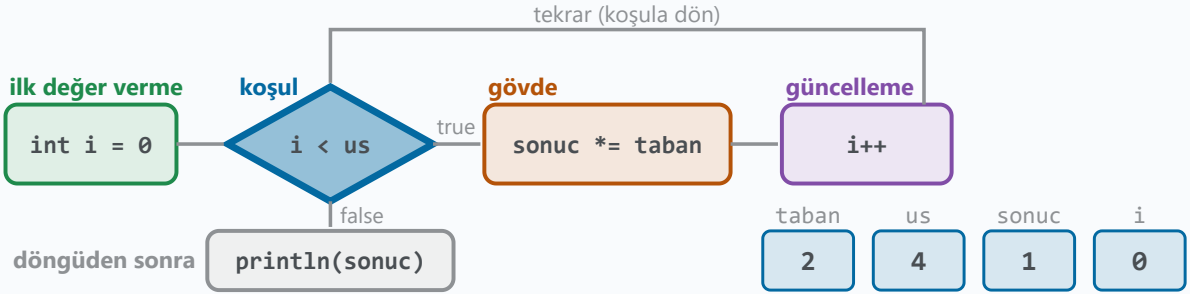
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
▶ for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

taban = 2 | us = 4 | sonuc = 1 | i = 0

ÇIKTI

Kod İzleme: 2^4



Koşul: 0 < 4 → **true**.

Adım 4 / 17

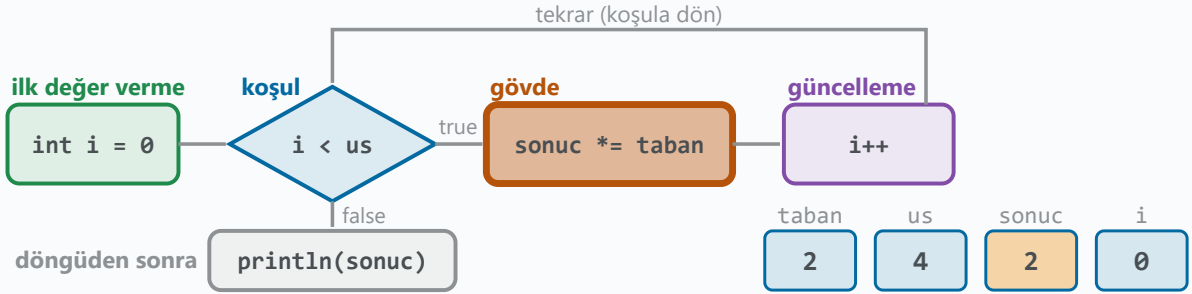
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
▶ for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

`taban = 2` `us = 4` `sonuc = 1` `i = 0`

ÇIKTI

Kod İzleme: 2^4



sonuc = 1 × 2 = **2** (= 2^1).

Adım 5 / 17

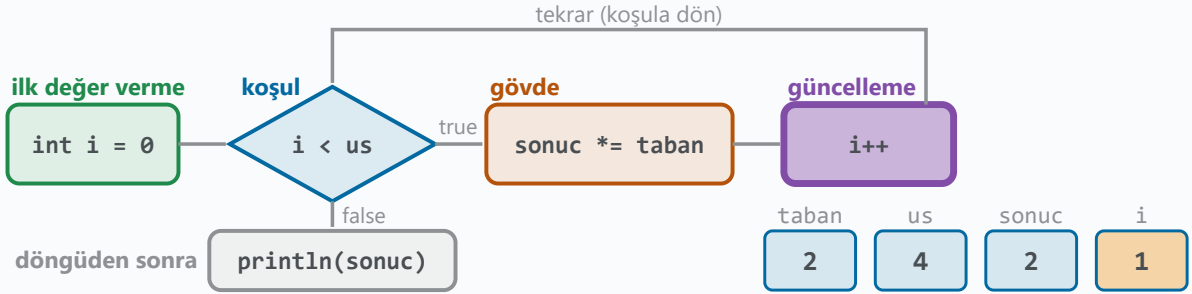
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
3 for (int i = 0; i < us; i++) {
▶     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

`taban = 2` `us = 4` `sonuc = 2` `i = 0`

ÇIKTI

Kod İzleme: 2^4



Güncelleme: i = 1.

Adım 6 / 17

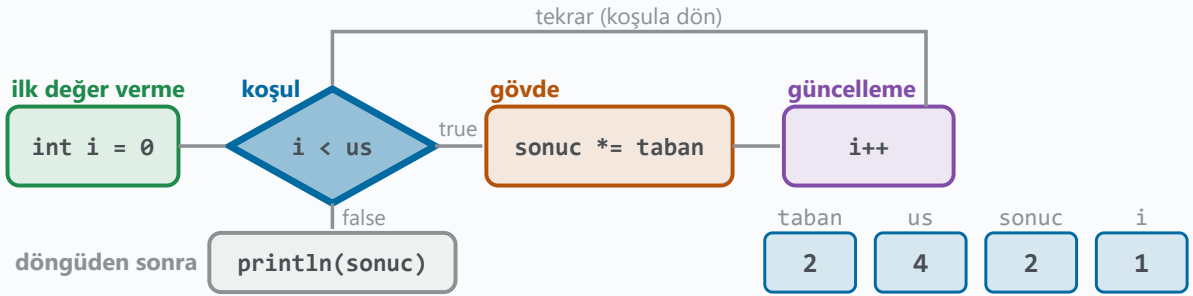
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
▶ for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

taban = 2 | us = 4 | sonuc = 2 | i = 1

ÇIKTI

Kod İzleme: 2^4



Koşul: 1 < 4 → **true**.

Adım 7 / 17

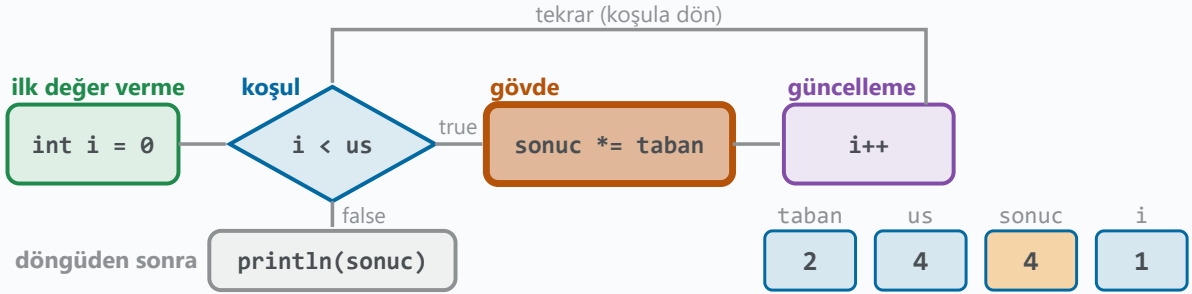
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
▶ for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

`taban = 2` `us = 4` `sonuc = 2` `i = 1`

ÇIKTI

Kod İzleme: 2^4



sonuc = 2 × 2 = **4** (= 2^2).

Adım 8 / 17

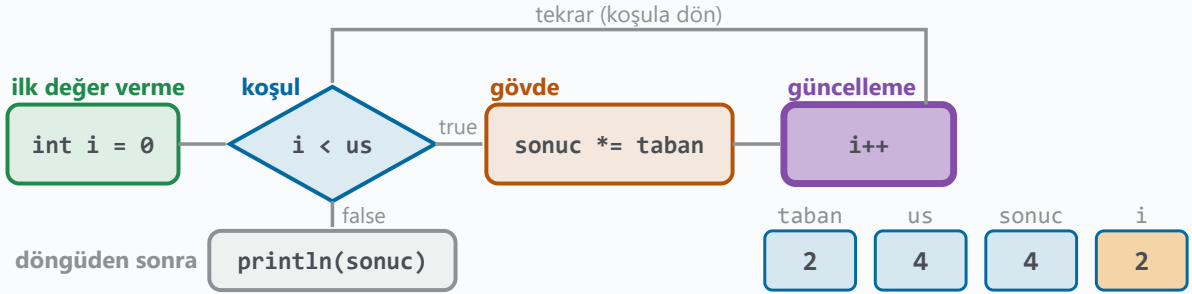
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
3 for (int i = 0; i < us; i++) {
▶     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

`taban = 2` `us = 4` `sonuc = 4` `i = 1`

ÇIKTI

Kod İzleme: 2^4



Güncelleme: i = 2.

Adım 9 / 17

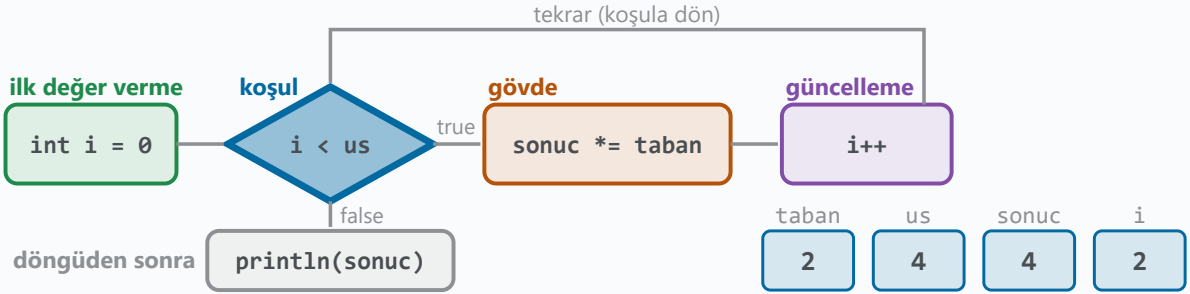
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
▶ for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

taban = 2 | us = 4 | sonuc = 4 | i = 2

ÇIKTI

Kod İzleme: 2^4



Koşul: 2 < 4 → **true**.

Adım 10 / 17

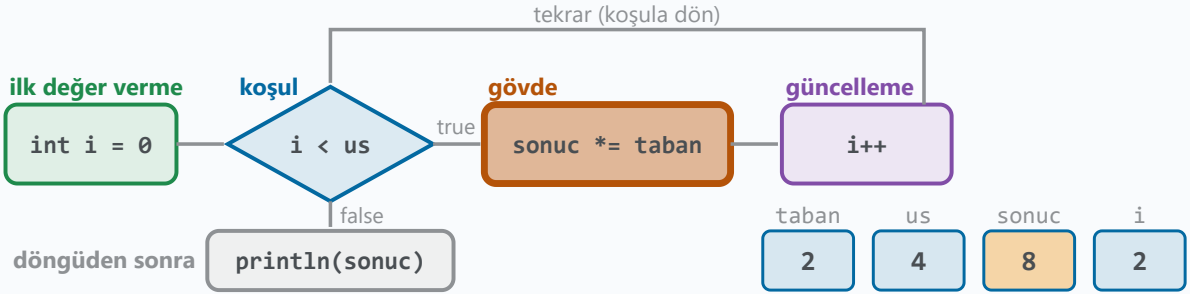
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
▶ for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

`taban = 2` `us = 4` `sonuc = 4` `i = 2`

ÇIKTI

Kod İzleme: 2^4



sonuc = 4 × 2 = **8** (= 2^3).

Adım 11 / 17

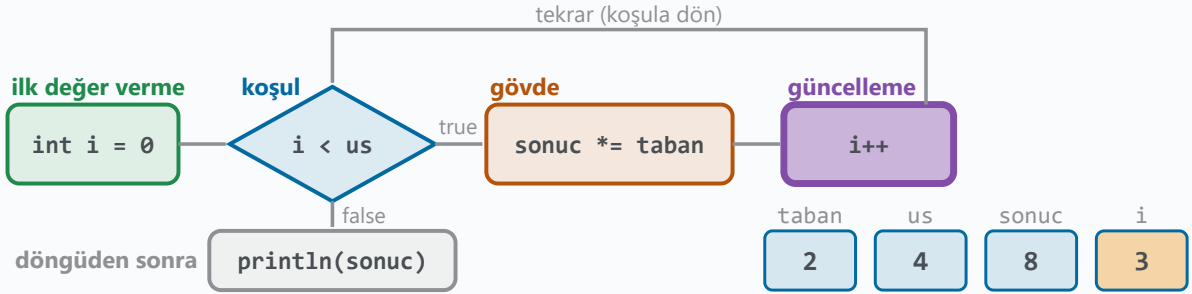
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
3 for (int i = 0; i < us; i++) {
▶     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

`taban = 2` `us = 4` `sonuc = 8` `i = 2`

ÇIKTI

Kod İzleme: 2^4



Güncelleme: i = 3.

Adım 12 / 17

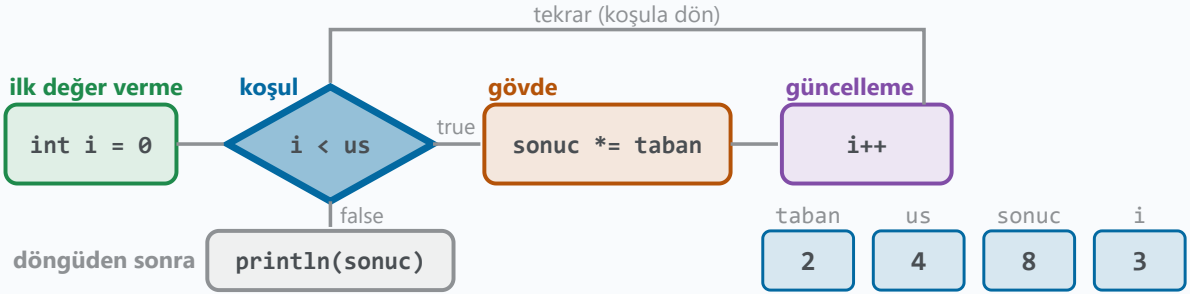
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
▶ for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

taban = 2 | us = 4 | sonuc = 8 | i = 3

ÇIKTI

Kod İzleme: 2^4



Koşul: 3 < 4 → **true**.

Adım 13 / 17

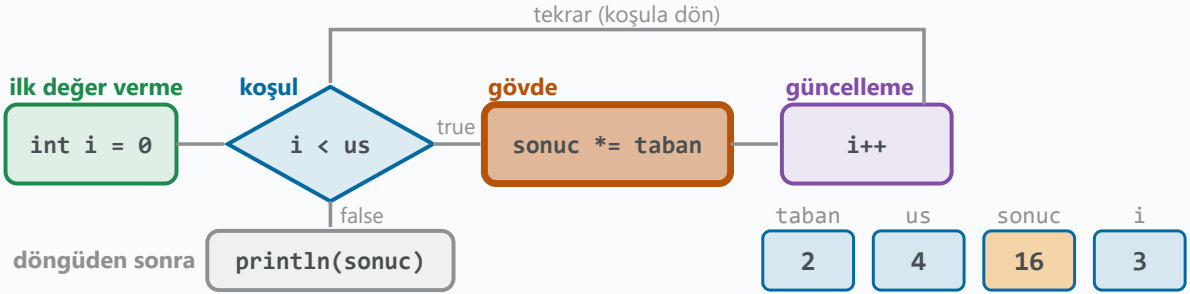
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
▶ for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

`taban = 2` `us = 4` `sonuc = 8` `i = 3`

ÇIKTI

Kod İzleme: 2^4



sonuc = 8 × 2 = **16** (= 2^4).

Adım 14 / 17

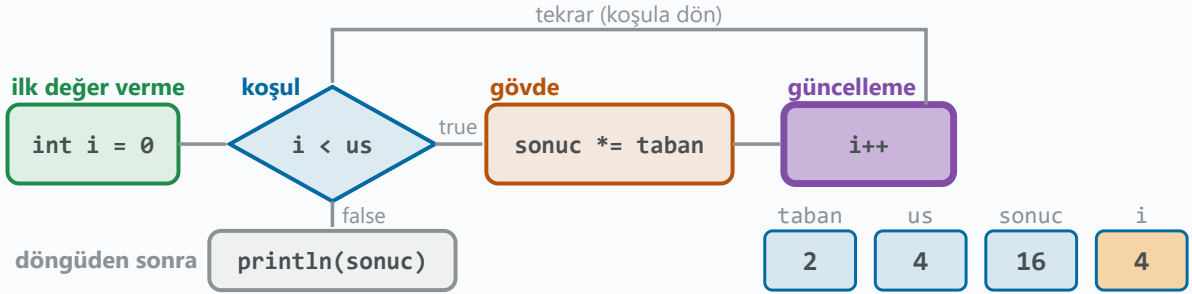
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
3 for (int i = 0; i < us; i++) {
▶     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

`taban = 2` `us = 4` `sonuc = 16` `i = 3`

ÇIKTI

Kod İzleme: 2^4



Güncelleme: i = 4.

Adım 15 / 17

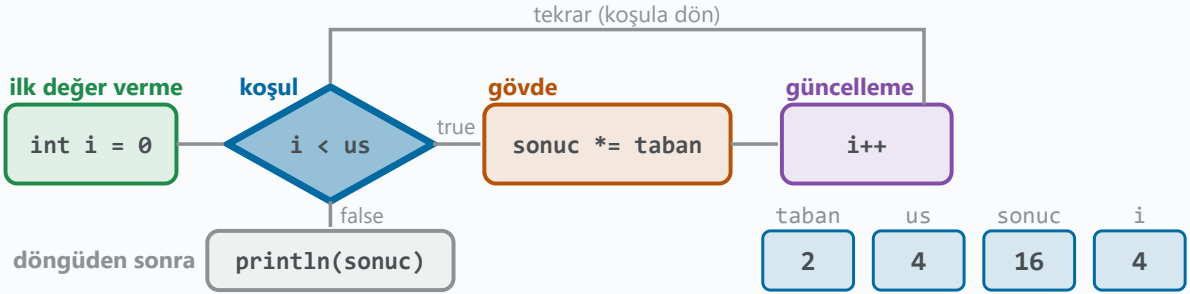
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
▶ for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

taban = 2 | us = 4 | sonuc = 16 | i = 4

ÇIKTI

Kod İzleme: 2^4



Koşul: 4 < 4 → **false**.

Adım 16 / 17

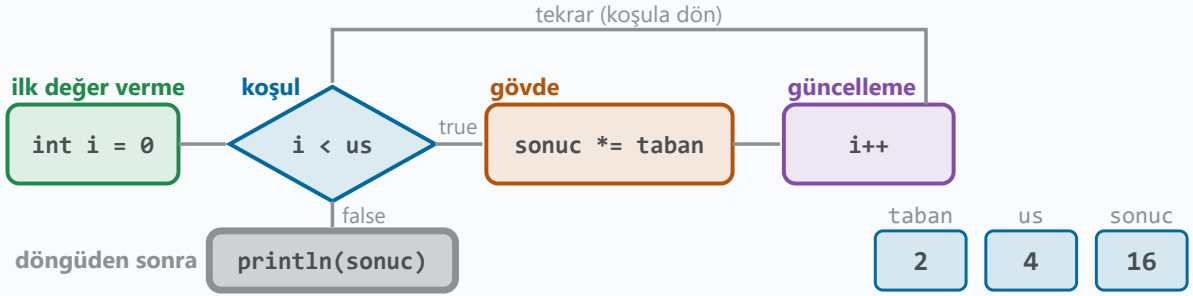
```
1 int taban = 2, us = 4;
2 int sonuc = 1;
▶ for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
6 System.out.println(sonuc);
```

DEĞİŞKENLER

`taban = 2` `us = 4` `sonuc = 16` `i = 4`

ÇIKTI

Kod İzleme: 2^4



16 yazılır. Değişmez: her turun başında sonuc = 2^i.

Adım 17 / 17

```
1 int taban = 2, us = 4;
2 int sonuc = 1;
3 for (int i = 0; i < us; i++) {
4     sonuc *= taban;
5 }
▶ System.out.println(sonuc);
```

DEĞİŞKENLER

taban = 2 | us = 4 | sonuc = 16

ÇIKTI

```
16
```

ÖRNEK

Asal Sayı

```
int n = 29;
boolean asal = n >= 2;
for (int i = 2; asal && i <= n / i; i++) {
    if (n % i == 0) asal = false;
}
System.out.println(asal);
```



- 0 ve 1 asal değildir: `asal = n >= 2`.
- Bölen aramak için √n'ye kadar gitmek yeter: `i <= n / i` ≡ i × i ≤ n (taşma riski yok).
- `asal &&` sayesinde bölen bulununca döngü durur.

Kod İzleme: 29 asal mı?



n = 29 asal mı?

Adım 1 / 17

```
▶ int n = 29;
2 boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

n = 29

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: 29 asal mı?



29 >= 2 → **true**. Aksi kanıtlanana kadar asal kabul ediyoruz.

Adım 2 / 17

```
1 int n = 29;
▶ boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

n = 29 asal = true

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: 29 asal mı?



Başlangıç: ilk bölen adayı i = 2.

Adım 3 / 17

```
1 int n = 29;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

n = 29 | asal = true | i = 2

ÇIKTI

Kod İzleme: 29 asal mı?



Koşul: true && 2 <= 29/2 = 14 → **true**.

Adım 4 / 17

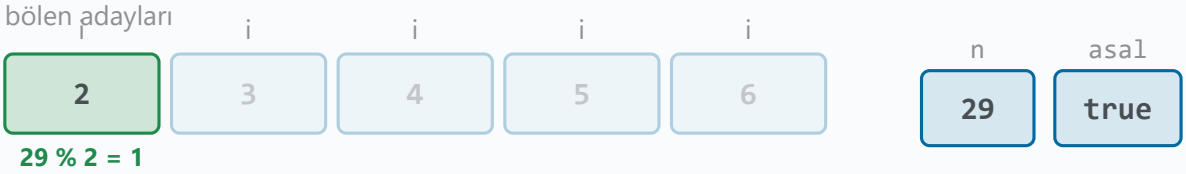
```
1 int n = 29;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 29` `asal = true` `i = 2`

ÇIKTI

Kod İzleme: 29 asal mı?



29 % 2 = 1 → bölmüyor.

Adım 5 / 17

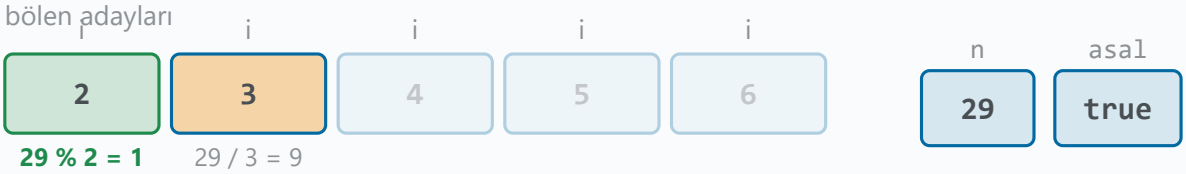
```
1 int n = 29;
2 boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
▶     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 29` `asal = true` `i = 2`

ÇIKTI

Kod İzleme: 29 asal mı?



Güncelleme: i = 3.

Adım 6 / 17

```
1 int n = 29;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 29` `asal = true` `i = 3`

ÇIKTI

Kod İzleme: 29 asal mı?



Koşul: true && 3 <= 29/3 = 9 → **true**.

Adım 7 / 17

```
1 int n = 29;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

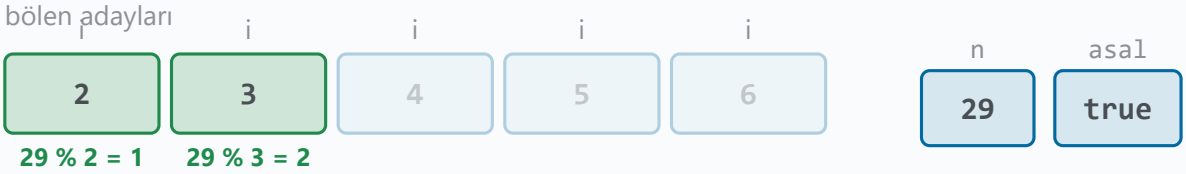
DEĞİŞKENLER

n = **29** | asal = **true** | i = **3**

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: 29 asal mı?



29 % 3 = 2 → bölmüyor.

Adım 8 / 17

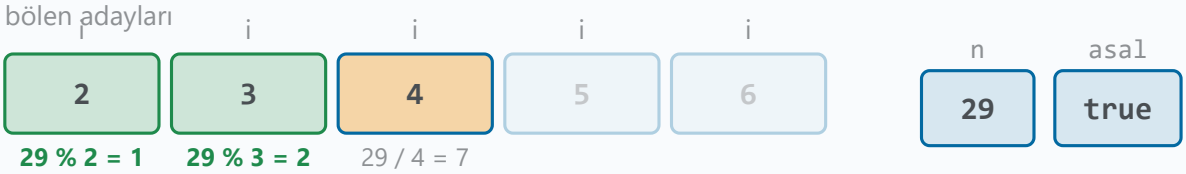
```
1 int n = 29;
2 boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
▶     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

n = 29 | asal = true | i = 3

ÇIKTI

Kod İzleme: 29 asal mı?



Güncelleme: i = 4.

Adım 9 / 17

```
1 int n = 29;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 29` `asal = true` `i = 4`

ÇIKTI

Kod İzleme: 29 asal mı?



Koşul: true && 4 <= 29/4 = 7 → **true**.

Adım 10 / 17

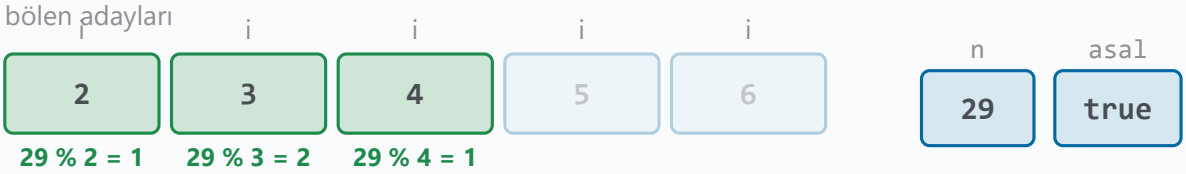
```
1 int n = 29;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 29` `asal = true` `i = 4`

ÇIKTI

Kod İzleme: 29 asal mı?



29 % 4 = 1 → bölmüyor.

Adım 11 / 17

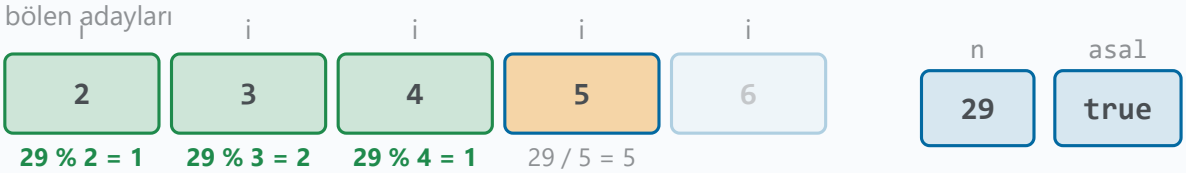
```
1 int n = 29;
2 boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
▶     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 29` `asal = true` `i = 4`

ÇIKTI

Kod İzleme: 29 asal mı?



Güncelleme: i = 5.

Adım 12 / 17

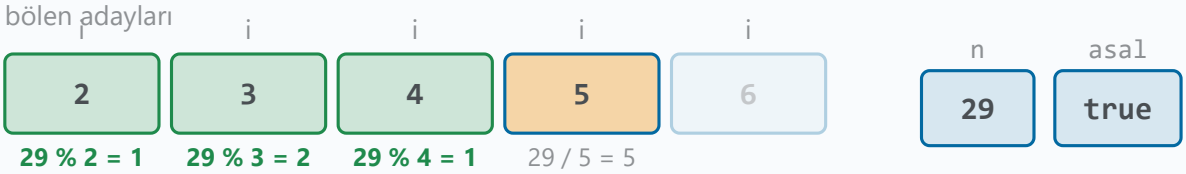
```
1 int n = 29;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 29` `asal = true` `i = 5`

ÇIKTI

Kod İzleme: 29 asal mı?



Koşul: true && 5 <= 29/5 = 5 → **true**.

Adım 13 / 17

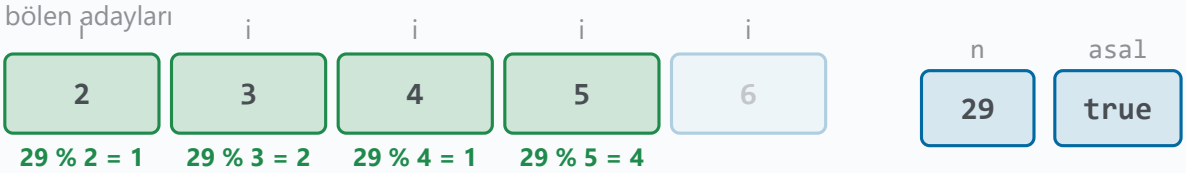
```
1 int n = 29;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 29` `asal = true` `i = 5`

ÇIKTI

Kod İzleme: 29 asal mı?



29 % 5 = 4 → bölmüyor.

Adım 14 / 17

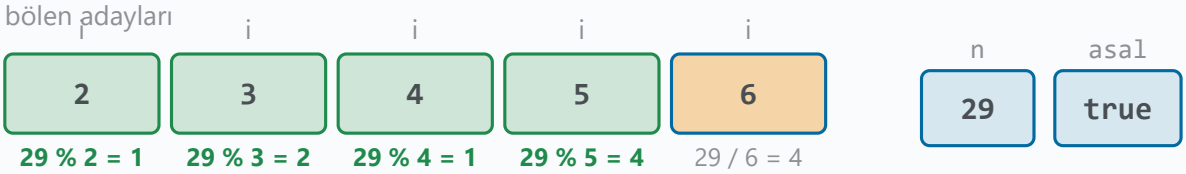
```
1 int n = 29;
2 boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
▶     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 29` `asal = true` `i = 5`

ÇIKTI

Kod İzleme: 29 asal mı?



Güncelleme: i = 6.

Adım 15 / 17

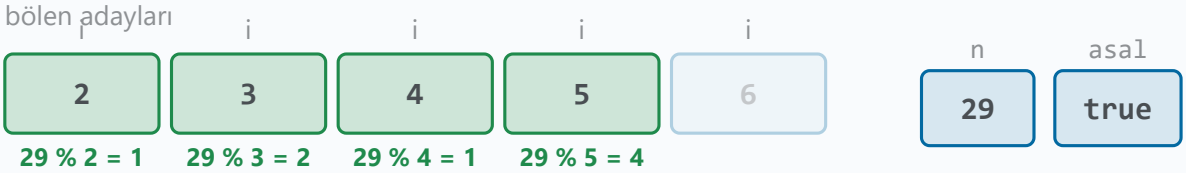
```
1 int n = 29;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 29` `asal = true` `i = 6`

ÇIKTI

Kod İzleme: 29 asal mı?



Koşul: true && 6 <= 29/6 = 4 → **false**: √29'u geçtik, daha büyük bölen aramaya gerek yok.

Adım 16 / 17

```
1 int n = 29;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

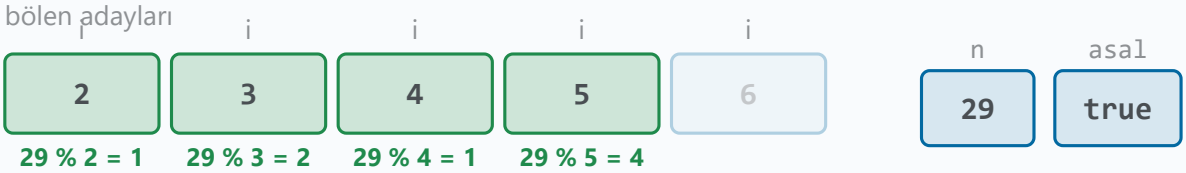
DEĞİŞKENLER

`n = 29` `asal = true` `i = 6`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: 29 asal mı?



true yazılır. 4 bölen denendi.

Adım 17 / 17

```
1 int n = 29;
2 boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
▶ System.out.println(asal);
```

DEĞİŞKENLER

n = 29 | asal = true

ÇIKTI

```
true
```

Kod İzleme (devam): 21 asal mı?



n = 21 asal mı?

Adım 1 / 11

```
▶ int n = 21;
2 boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

n = **21**

ÇIKTI

Kod İzleme (devam): 21 asal mı?



21 >= 2 → **true**. Aksi kanıtlanana kadar asal kabul ediyoruz.

Adım 2 / 11

```
1 int n = 21;
▶ boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 21` `asal = true`

ÇIKTI

Kod İzleme (devam): 21 asal mı?



Başlangıç: ilk bölen adayı i = 2.

Adım 3 / 11

```
1 int n = 21;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 21` `asal = true` `i = 2`

ÇIKTI

Kod İzleme (devam): 21 asal mı?



Koşul: true && 2 <= 21/2 = 10 → **true**.

Adım 4 / 11

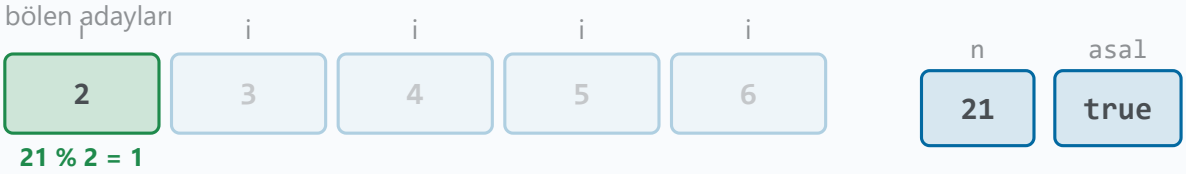
```
1 int n = 21;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 21` `asal = true` `i = 2`

ÇIKTI

Kod İzleme (devam): 21 asal mı?



21 % 2 = 1 → bölmüyor.

Adım 5 / 11

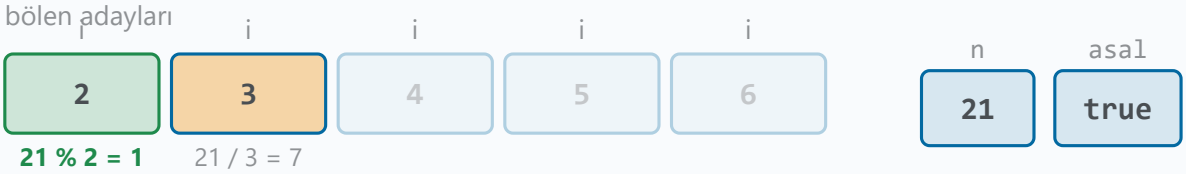
```
1 int n = 21;
2 boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
▶     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 21` `asal = true` `i = 2`

ÇIKTI

Kod İzleme (devam): 21 asal mı?



Güncelleme: i = 3.

Adım 6 / 11

```
1 int n = 21;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

n = 21 | asal = true | i = 3

ÇIKTI

Kod İzleme (devam): 21 asal mı?



Koşul: true && 3 <= 21/3 = 7 → **true**.

Adım 7 / 11

```
1 int n = 21;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

n = **21** | asal = **true** | i = **3**

ÇIKTI

Kod İzleme (devam): 21 asal mı?



21 % 3 = 0 → **bölen bulundu**: asal = false.

Adım 8 / 11

```
1 int n = 21;
2 boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
▶     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 21` `asal = false` `i = 3`

ÇIKTI

Kod İzleme (devam): 21 asal mı?



Güncelleme: i = 4.

Adım 9 / 11

```
1 int n = 21;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

n = 21 | asal = false | i = 4

ÇIKTI

Kod İzleme (devam): 21 asal mı?



Koşul: asal = false → && kısa devre yapar, `i <= n / i` hiç hesaplanmaz → döngü biter.

Adım 10 / 11

```
1 int n = 21;
2 boolean asal = n >= 2;
▶ for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
6 System.out.println(asal);
```

DEĞİŞKENLER

`n = 21` `asal = false` `i = 4`

ÇIKTI

Kod İzleme (devam): 21 asal mı?



false yazılır. Bölen bulunur bulunmaz döngü durdu.

Adım 11 / 11

```
1 int n = 21;
2 boolean asal = n >= 2;
3 for (int i = 2; asal && i <= n / i; i++) {
4     if (n % i == 0) asal = false;
5 }
▶ System.out.println(asal);
```

DEĞİŞKENLER

`n = 21` `asal = false`

ÇIKTI

```
false
```

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

2’den 20’ye her k için asal testi (iç döngü) yapılır.

Adım 1 / 21

```
▶ int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
4     for (int i = 2; i <= k / i; i++) {
5         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20

ÇIKTI

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

k = 2: iç döngü hiç çalışmaz (2 > k / 2) → asal = true → yazılır.

Adım 2 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
▶     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
5         if (k % i == 0) { asal = false; break; }
6     }
▶     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 k = 2 asal = true

ÇIKTI

```
2
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 3: iç döngü hiç çalışmaz (2 > k / 2) → asal = true → yazılır.

Adım 3 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
▶     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
5         if (k % i == 0) { asal = false; break; }
6     }
▶     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

`n = 20` `k = 3` `asal = true`

ÇIKTI

```
2 3
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 4: 4 % 2 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 4 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

`n = 20` `k = 4` `asal = false` `i = 2`

ÇIKTI

```
2 3
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

k = 5: 2 … 2 arasında bölen yok → asal = true → yazılır.

Adım 5 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
      boolean asal = true;
      for (int i = 2; i <= k / i; i++) {
5         if (k % i == 0) { asal = false; break; }
6     }
      if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 | k = 5 | asal = true

ÇIKTI

```
2 3 5
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 6: 6 % 2 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 6 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

`n = 20` `k = 6` `asal = false` `i = 2`

ÇIKTI

```
2 3 5
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

k = 7: 2 … 2 arasında bölen yok → asal = true → yazılır.

Adım 7 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
      boolean asal = true;
      for (int i = 2; i <= k / i; i++) {
5         if (k % i == 0) { asal = false; break; }
6     }
      if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 | k = 7 | asal = true

ÇIKTI

```
2 3 5 7
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 8: 8 % 2 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 8 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 | k = 8 | asal = false | i = 2

ÇIKTI

```
2 3 5 7
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 9: 9 % 3 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 9 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 | k = 9 | asal = false | i = 3

ÇIKTI

```
2 3 5 7
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 10: 10 % 2 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 10 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

`n = 20` `k = 10` `asal = false` `i = 2`

ÇIKTI

```
2 3 5 7
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 11: 2 … 3 arasında bölen yok → asal = true → yazılır.

Adım 11 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
      boolean asal = true;
      for (int i = 2; i <= k / i; i++) {
5         if (k % i == 0) { asal = false; break; }
6     }
      if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 | k = 11 | asal = true

ÇIKTI

```
2 3 5 7 11
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 12: 12 % 2 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 12 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

`n = 20` `k = 12` `asal = false` `i = 2`

ÇIKTI

```
2 3 5 7 11
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 13: 2 … 3 arasında bölen yok → asal = true → yazılır.

Adım 13 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
      boolean asal = true;
      for (int i = 2; i <= k / i; i++) {
5         if (k % i == 0) { asal = false; break; }
6     }
      if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 | k = 13 | asal = true

ÇIKTI

```
2 3 5 7 11 13
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 14: 14 % 2 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 14 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

`n = 20` `k = 14` `asal = false` `i = 2`

ÇIKTI

```
2 3 5 7 11 13
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 15: 15 % 3 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 15 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

`n = 20` `k = 15` `asal = false` `i = 3`

ÇIKTI

```
2 3 5 7 11 13
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 16: 16 % 2 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 16 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

`n = 20` `k = 16` `asal = false` `i = 2`

ÇIKTI

```
2 3 5 7 11 13
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 17: 2 … 4 arasında bölen yok → asal = true → yazılır.

Adım 17 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
      boolean asal = true;
      for (int i = 2; i <= k / i; i++) {
5         if (k % i == 0) { asal = false; break; }
6     }
      if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 | k = 17 | asal = true

ÇIKTI

```
2 3 5 7 11 13 17
```

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 18: 18 % 2 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 18 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 | k = 18 | asal = false | i = 2

ÇIKTI

```
2 3 5 7 11 13 17
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 19: 2 … 4 arasında bölen yok → asal = true → yazılır.

Adım 19 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
      boolean asal = true;
      for (int i = 2; i <= k / i; i++) {
5         if (k % i == 0) { asal = false; break; }
6     }
      if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 | k = 19 | asal = true

ÇIKTI

```
2 3 5 7 11 13 17 19
```

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

k = 20: 20 % 2 = 0 → asal = false, **break** (iç döngü erken biter).

Adım 20 / 21

```
1 int n = 20;
2 for (int k = 2; k <= n; k++) {
3     boolean asal = true;
▶     for (int i = 2; i <= k / i; i++) {
▶         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = 20 | k = 20 | asal = false | i = 2

ÇIKTI

```
2 3 5 7 11 13 17 19
```

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

1’den n’ye Asal Sayıları Listeleme (n = 20)

yeşil = asal, soluk = bileşik, kesikli = henüz bakılmadı



k = 21 > 20: bitti. 2..20 arasında **8** asal var. (Eratosthenes eleği daha hızlıdır.)

Adım 21 / 21

```
1 int n = 20;
▶ for (int k = 2; k <= n; k++) {
3     boolean asal = true;
4     for (int i = 2; i <= k / i; i++) {
5         if (k % i == 0) { asal = false; break; }
6     }
7     if (asal) System.out.print(k + " ");
8 }
```

DEĞİŞKENLER

n = **20**

ÇIKTI

```
2 3 5 7 11 13 17 19
```

ÖRNEK

Fibonacci Dizisi

```
int adet = 5;
long a = 0, b = 1;
for (int i = 0; i < adet; i++) {
    System.out.print(a + " ");
    long sonraki = a + b;
    a = b;
    b = sonraki;
}
```



çıktı: 0 1 1 2 3

- Her terim önceki ikisinin toplamı: 0 1 1 2 3 5 8 …
- Güncelleme sırası iki önceki değeri korur; `b = sonraki` önce yazılsaydı eski b kaybolurdu.
- Dizi hızlı büyür; long bile 93. terimden sonra taşar.

Kod İzleme: ilk 5 Fibonacci sayısı



İlk **5** Fibonacci sayısı yazılacak.

Adım 1 / 18

```
▶ int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
5     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

`adet = 5`

ÇIKTI

Kod İzleme: ilk 5 Fibonacci sayısı



Dizinin ilk iki terimi: a = 0, b = 1.

Adım 2 / 18

```
1 int adet = 5;
▶ long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
5     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

adet = 5 | a = 0 | b = 1

ÇIKTI

Kod İzleme: ilk 5 Fibonacci sayısı



i = 0 < 5 → true. Sıradaki terim **a = 0** yazılır.


```
1 int adet = 5;
2 long a = 0, b = 1;
▶ for (int i = 0; i < adet; i++) {
▶     System.out.print(a + " ");
5     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

adet = 5 | a = 0 | b = 1 | i = 0

ÇIKTI

```
0
```

Kod İzleme: ilk 5 Fibonacci sayısı



sonraki = 0 + 1 = **1**.

Adım 4 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
▶     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

`adet = 5` `a = 0` `b = 1` `i = 0` `sonraki = 1`

ÇIKTI

```
0
```

Kod İzleme: ilk 5 Fibonacci sayısı



Kaydırma: a = 1, b = 1. Sıra önemli: önce a = b, sonra b = sonraki. Sonra i++.

Adım 5 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
5     long sonraki = a + b;
▶     a = b;
▶     b = sonraki;
8 }
```

DEĞİŞKENLER

`adet = 5` `a = 1` `b = 1` `i = 0` `sonraki = 1`

ÇIKTI

```
0
```

Kod İzleme: ilk 5 Fibonacci sayısı



i = 1 < 5 → true. Sıradaki terim **a = 1** yazılır.

Adım 6 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
▶ for (int i = 0; i < adet; i++) {
▶     System.out.print(a + " ");
5     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

adet = **5** | a = **1** | b = **1** | i = **1**

ÇIKTI

```
0 1
```

Kod İzleme: ilk 5 Fibonacci sayısı



sonraki = 1 + 1 = **2**.

Adım 7 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
▶     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

`adet = 5` `a = 1` `b = 1` `i = 1` `sonraki = 2`

ÇIKTI

```
0 1
```

Kod İzleme: ilk 5 Fibonacci sayısı



Kaydırma: a = 1, b = 2. Sıra önemli: önce a = b, sonra b = sonraki. Sonra i++.

Adım 8 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
5     long sonraki = a + b;
▶     a = b;
▶     b = sonraki;
8 }
```

DEĞİŞKENLER

adet = 5 | a = 1 | b = 2 | i = 1 | sonraki = 2

ÇIKTI

```
0 1
```

Kod İzleme: ilk 5 Fibonacci sayısı



i = 2 < 5 → true. Sıradaki terim **a = 1** yazılır.

Adım 9 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
▶ for (int i = 0; i < adet; i++) {
▶     System.out.print(a + " ");
5     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

adet = **5** | a = **1** | b = **2** | i = **2**

ÇIKTI

```
0 1 1
```

Kod İzleme: ilk 5 Fibonacci sayısı



sonraki = 1 + 2 = **3**.

Adım 10 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
▶     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

`adet = 5` `a = 1` `b = 2` `i = 2` `sonraki = 3`

ÇIKTI

```
0 1 1
```

Kod İzleme: ilk 5 Fibonacci sayısı



Kaydırma: a = 2, b = 3. Sıra önemli: önce a = b, sonra b = sonraki. Sonra i++.

Adım 11 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
5     long sonraki = a + b;
▶     a = b;
▶     b = sonraki;
8 }
```

DEĞİŞKENLER

`adet = 5` `a = 2` `b = 3` `i = 2` `sonraki = 3`

ÇIKTI

```
0 1 1
```

Kod İzleme: ilk 5 Fibonacci sayısı



i = 3 < 5 → true. Sıradaki terim **a = 2** yazılır.

Adım 12 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
▶ for (int i = 0; i < adet; i++) {
▶     System.out.print(a + " ");
5     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

adet = **5** | a = **2** | b = **3** | i = **3**

ÇIKTI

```
0 1 1 2
```

Kod İzleme: ilk 5 Fibonacci sayısı



sonraki = 2 + 3 = **5**.

Adım 13 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
▶     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

`adet = 5` `a = 2` `b = 3` `i = 3` `sonraki = 5`

ÇIKTI

```
0 1 1 2
```

Kod İzleme: ilk 5 Fibonacci sayısı



Kaydırma: a = 3, b = 5. Sıra önemli: önce a = b, sonra b = sonraki. Sonra i++.

Adım 14 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
5     long sonraki = a + b;
▶     a = b;
▶     b = sonraki;
8 }
```

DEĞİŞKENLER

`adet = 5` `a = 3` `b = 5` `i = 3` `sonraki = 5`

ÇIKTI

```
0 1 1 2
```

Kod İzleme: ilk 5 Fibonacci sayısı



i = 4 < 5 → true. Sıradaki terim **a = 3** yazılır.


```
1 int adet = 5;
2 long a = 0, b = 1;
▶ for (int i = 0; i < adet; i++) {
▶     System.out.print(a + " ");
5     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

adet = **5** | a = **3** | b = **5** | i = **4**

ÇIKTI

```
0 1 1 2 3
```

Kod İzleme: ilk 5 Fibonacci sayısı



sonraki = 3 + 5 = **8**.

Adım 16 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
▶     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

`adet = 5` `a = 3` `b = 5` `i = 4` `sonraki = 8`

ÇIKTI

```
0 1 1 2 3
```

Kod İzleme: ilk 5 Fibonacci sayısı



Kaydırma: a = 5, b = 8. Sıra önemli: önce a = b, sonra b = sonraki. Sonra i++.

Adım 17 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
3 for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
5     long sonraki = a + b;
▶     a = b;
▶     b = sonraki;
8 }
```

DEĞİŞKENLER

`adet = 5` `a = 5` `b = 8` `i = 4` `sonraki = 8`

ÇIKTI

```
0 1 1 2 3
```

Kod İzleme: ilk 5 Fibonacci sayısı



Koşul: 5 < 5 → **false** → döngü biter. Çıktı: 0 1 1 2 3.

Adım 18 / 18

```
1 int adet = 5;
2 long a = 0, b = 1;
▶ for (int i = 0; i < adet; i++) {
4     System.out.print(a + " ");
5     long sonraki = a + b;
6     a = b;
7     b = sonraki;
8 }
```

DEĞİŞKENLER

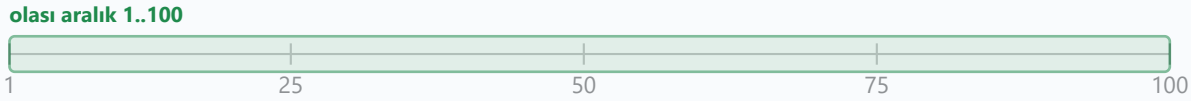
adet = **5** | a = **5** | b = **8** | i = **5**

ÇIKTI

```
0 1 1 2 3
```

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Sayı Tahmin Oyunu (while + Random + Scanner)



`Random` rastgele sayı üretir (dosya başında `import java.util.Random;` ve `java.util.Scanner`).

Adım 1 / 15

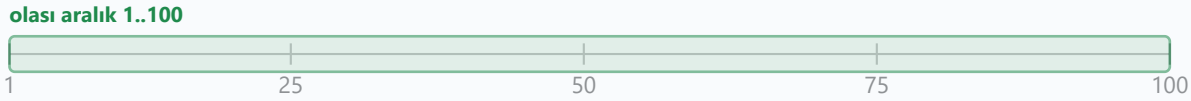
```
▶ Random rnd = new Random();
2 Scanner klavye = new Scanner(System.in);
3 int hedef = rnd.nextInt(100) + 1;  // 1..100
4 int tahmin = 0, deneme = 0;
5 while (tahmin != hedef) {
6   tahmin = klavye.nextInt();
7   deneme++;
8   if (tahmin < hedef) System.out.println("Yukarı");
9   if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

DEĞİŞKENLER

`rnd` = **Random**

ÇIKTI

Sayı Tahmin Oyunu (while + Random + Scanner)



Klavyeden okumak için Scanner.

Adım 2 / 15

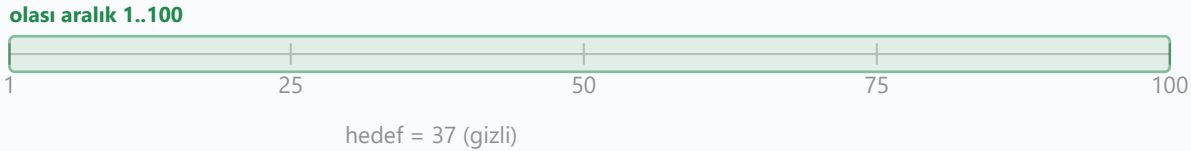
```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
5  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

DEĞİŞKENLER

`rnd = Random` `klavye = Scanner`

ÇIKTI

Sayı Tahmin Oyunu (while + Random + Scanner)



`nextInt(100)` 0..99 verir, +1 ile 1..100. Bu çalıştırmada **hedef = 37** çıktı (her çalıştırmada farklı).

Adım 3 / 15

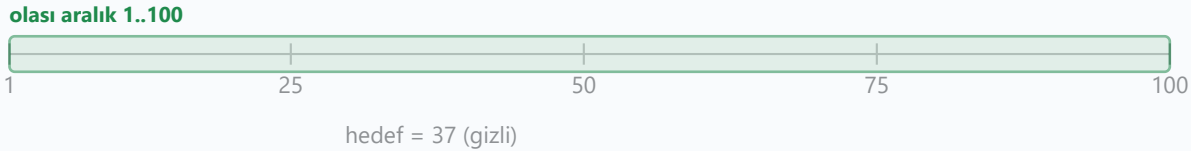
```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
▶  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
5  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

DEĞİŞKENLER

`hedef = 37`

ÇIKTI

Sayı Tahmin Oyunu (while + Random + Scanner)



hedef = 37 (gizli)

tahmin = 0 hiçbir zaman hedef olamaz → döngüye kesin girilir.

Adım 4 / 15

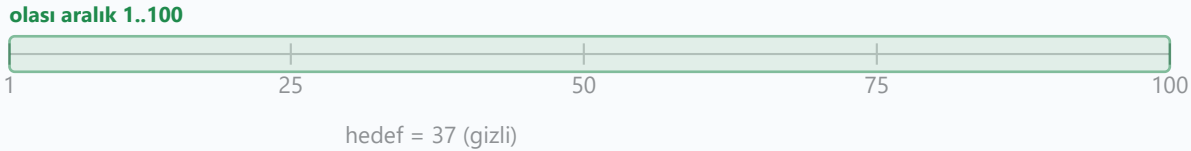
```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
▶ 4  int tahmin = 0, deneme = 0;
5  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

DEĞİŞKENLER

`hedef = 37` `tahmin = 0` `deneme = 0`

ÇIKTI

Sayı Tahmin Oyunu (while + Random + Scanner)



Koşul: 0 != 37 → **true**.

Adım 5 / 15

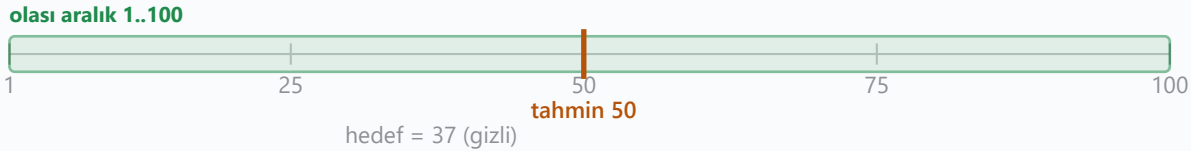
```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
▶  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

DEĞİŞKENLER

`hedef = 37` `tahmin = 0` `deneme = 0`

ÇIKTI

Sayı Tahmin Oyunu (while + Random + Scanner)



Kullanıcı **50** yazar; deneme = 1.

Adım 6 / 15

```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
5  while (tahmin != hedef) {
▶    tahmin = klavye.nextInt();
▶    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

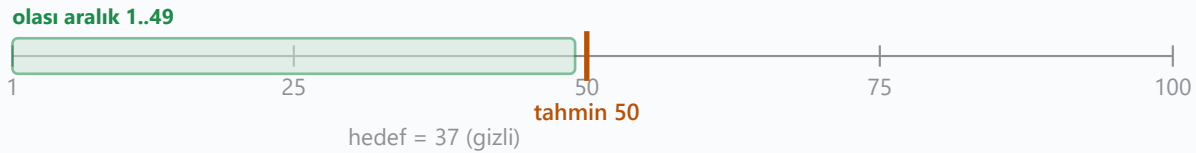
DEĞİŞKENLER

`hedef = 37` `tahmin = 50` `deneme = 1`

ÇIKTI

```
50
```

Sayı Tahmin Oyunu (while + Random + Scanner)



50 > 37 → «Aşağı». Olası aralık 1..49. Her seferinde ortayı söylemek aralığı yarıya indirir.

Adım 7 / 15

```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
5  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
▶    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

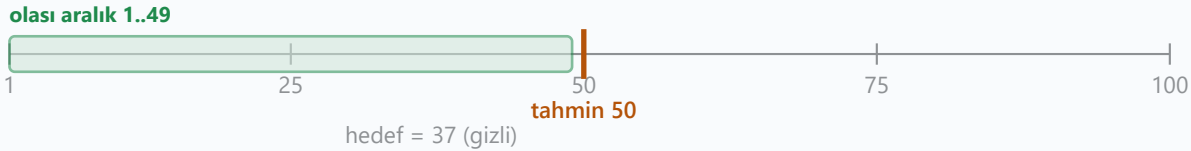
DEĞİŞKENLER

`hedef = 37` `tahmin = 50` `deneme = 1`

ÇIKTI

```
50
Aşağı
```

Sayı Tahmin Oyunu (while + Random + Scanner)



Koşul: 50 != 37 → **true**.

Adım 8 / 15

```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
▶  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

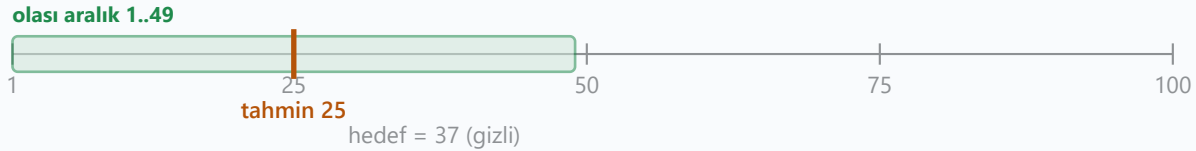
DEĞİŞKENLER

`hedef = 37` `tahmin = 50` `deneme = 1`

ÇIKTI

```
50
Aşağı
```

Sayı Tahmin Oyunu (while + Random + Scanner)



Kullanıcı **25** yazar; deneme = 2.

Adım 9 / 15

```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
5  while (tahmin != hedef) {
▶    tahmin = klavye.nextInt();
▶    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

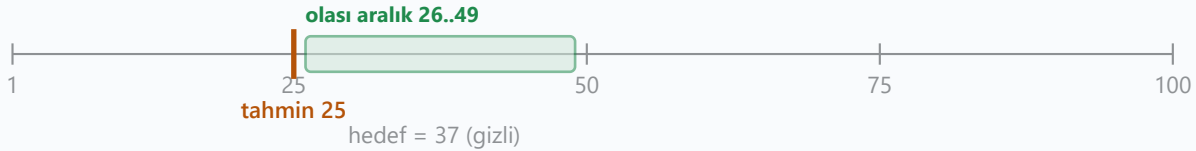
DEĞİŞKENLER

`hedef = 37` `tahmin = 25` `deneme = 2`

ÇIKTI

```
50
Aşağı
25
```

Sayı Tahmin Oyunu (while + Random + Scanner)



25 < 37 → «Yukarı». Olası aralık 26..49.

Adım 10 / 15

```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
5  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
▶    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

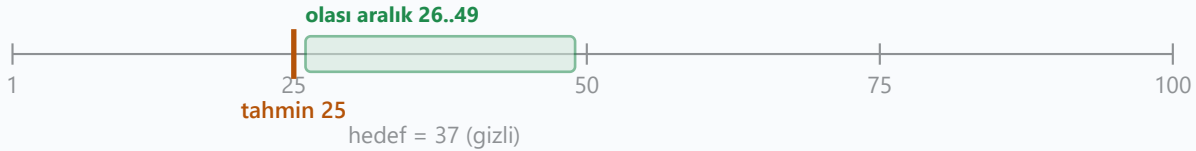
DEĞİŞKENLER

`hedef = 37` `tahmin = 25` `deneme = 2`

ÇIKTI

```
50
Aşağı
25
Yukarı
```

Sayı Tahmin Oyunu (while + Random + Scanner)



Koşul: 25 != 37 → **true**.

Adım 11 / 15

```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
▶  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

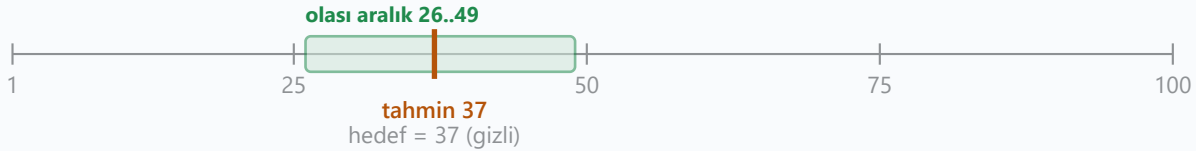
DEĞİŞKENLER

`hedef = 37` `tahmin = 25` `deneme = 2`

ÇIKTI

```
50
Aşağı
25
Yukarı
```

Sayı Tahmin Oyunu (while + Random + Scanner)



Kullanıcı **37** yazar; deneme = 3.

Adım 12 / 15

```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
5  while (tahmin != hedef) {
▶    tahmin = klavye.nextInt();
▶    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

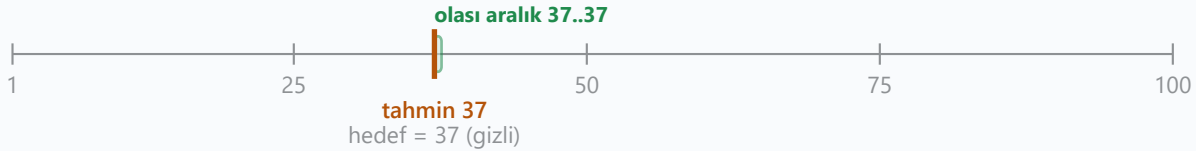
DEĞİŞKENLER

`hedef = 37` `tahmin = 37` `deneme = 3`

ÇIKTI

```
50
Aşağı
25
Yukarı
37
```

Sayı Tahmin Oyunu (while + Random + Scanner)



37 ne küçük ne büyük: iki if de false, ileti yok.

Adım 13 / 15

```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
5  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
▶    if (tahmin < hedef) System.out.println("Yukarı");
▶    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

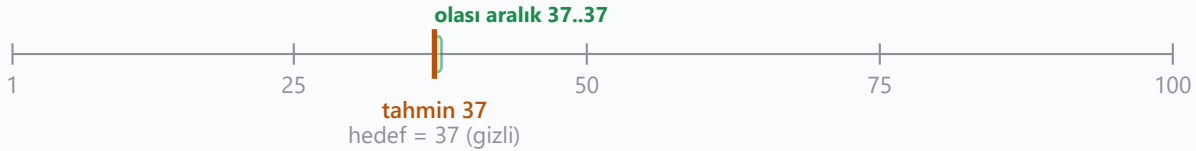
DEĞİŞKENLER

`hedef = 37` `tahmin = 37` `deneme = 3`

ÇIKTI

```
50
Aşağı
25
Yukarı
37
```

Sayı Tahmin Oyunu (while + Random + Scanner)



Koşul: 37 != 37 → **false** → döngü biter.


```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
▶  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
11 System.out.println(deneme + ". denemede bildin");
```

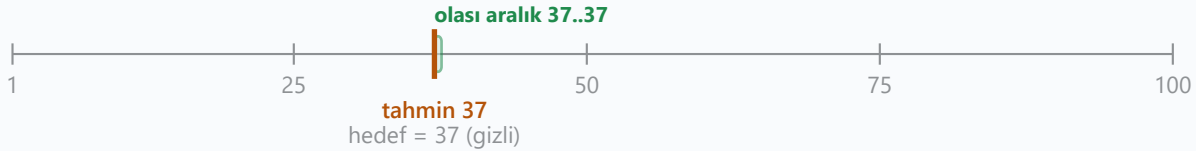
DEĞİŞKENLER

`hedef = 37` `tahmin = 37` `deneme = 3`

ÇIKTI

```
50
Aşağı
25
Yukarı
37
```

Sayı Tahmin Oyunu (while + Random + Scanner)



Kaç tur süreceği baştan belli değil → **while** doğru seçim.

Adım 15 / 15

```
1  Random rnd = new Random();
2  Scanner klavye = new Scanner(System.in);
3  int hedef = rnd.nextInt(100) + 1;  // 1..100
4  int tahmin = 0, deneme = 0;
5  while (tahmin != hedef) {
6    tahmin = klavye.nextInt();
7    deneme++;
8    if (tahmin < hedef) System.out.println("Yukarı");
9    if (tahmin > hedef) System.out.println("Aşağı");
10 }
▶  System.out.println(deneme + ". denemede bildin");
```

DEĞİŞKENLER

`hedef = 37` `tahmin = 37` `deneme = 3`

ÇIKTI

```
50
Aşağı
25
Yukarı
37
3. denemede bildin
```

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Bekçi Değer: `while (true)` + `break`, girdi 5, 3, 0

Girdi kuyruğu (klavyeden girilecek sayılar): **5, 3, 0**. Koşul `true` olduğundan döngü kendiliğinden bitmez; çıkış `break` ile sağlanır.

Adım 1 / 7

```
▶ Scanner sc = new Scanner(System.in);
▶ int toplam = 0;
3 while (true) {
4     int sayi = sc.nextInt();
5     if (sayi == 0) break;
6     toplam += sayi;
7 }
8 System.out.println("Toplam: " + toplam);
```

DEĞİŞKENLER

`toplam = 0` `girdi = [5, 3, 0]`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Bekçi Değer: `while (true)` + `break`, girdi 5, 3, 0

`nextInt()` kuyruktan **5** okur. 5 == 0 → false, break çalışmaz.

Adım 2 / 7

```
1 Scanner sc = new Scanner(System.in);
2 int toplam = 0;
3 while (true) {
▶     int sayi = sc.nextInt();
▶     if (sayi == 0) break;
6     toplam += sayi;
7 }
8 System.out.println("Toplam: " + toplam);
```

DEĞİŞKENLER

`toplam = 0` `sayi = 5`

`girdi = [3, 0]`

ÇIKTI

Bekçi Değer: `while (true)` + `break`, girdi 5, 3, 0

toplam = 0 + 5 = 5. Döngü başa döner (true).

Adım 3 / 7

```
1 Scanner sc = new Scanner(System.in);
2 int toplam = 0;
3 while (true) {
4     int sayi = sc.nextInt();
5     if (sayi == 0) break;
▶     toplam += sayi;
7 }
8 System.out.println("Toplam: " + toplam);
```

DEĞİŞKENLER

`toplam = 5` `sayi = 5`

`girdi = [3, 0]`

ÇIKTI

Bekçi Değer: `while (true)` + `break`, girdi 5, 3, 0

`nextInt()` → **3**. 3 == 0 → false.

Adım 4 / 7

```
1 Scanner sc = new Scanner(System.in);
2 int toplam = 0;
3 while (true) {
▶     int sayi = sc.nextInt();
▶     if (sayi == 0) break;
6     toplam += sayi;
7 }
8 System.out.println("Toplam: " + toplam);
```

DEĞİŞKENLER

`toplam = 5` `sayi = 3`

`girdi = [0]`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Bekçi Değer: `while (true)` + `break`, girdi 5, 3, 0

toplam = 5 + 3 = 8.

Adım 5 / 7

```
1 Scanner sc = new Scanner(System.in);
2 int toplam = 0;
3 while (true) {
4     int sayi = sc.nextInt();
5     if (sayi == 0) break;
▶     toplam += sayi;
7 }
8 System.out.println("Toplam: " + toplam);
```

DEĞİŞKENLER

`toplam = 8` `sayi = 3`

`girdi = [0]`

ÇIKTI

ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Bekçi Değer: `while (true)` + `break`, girdi 5, 3, 0

`nextInt()` → **0** (bekçi değer). 0 == 0 → true → `break` : döngüden çıkılır; 0 toplama eklenmez.

Adım 6 / 7

```
1 Scanner sc = new Scanner(System.in);
2 int toplam = 0;
3 while (true) {
▶     int sayi = sc.nextInt();
▶     if (sayi == 0) break;
6     toplam += sayi;
7 }
8 System.out.println("Toplam: " + toplam);
```

DEĞİŞKENLER

`toplam = 8` `sayi = 0`

`girdi = []`

ÇIKTI

Bekçi Değer: `while (true)` + `break`, girdi 5, 3, 0

`Toplam: 8` yazılır. Bekçi değer (sentinel) veri sonunu işaretler; kendisi veriye katılmaz.

Adım 7 / 7

```
1 Scanner sc = new Scanner(System.in);
2 int toplam = 0;
3 while (true) {
4     int sayi = sc.nextInt();
5     if (sayi == 0) break;
6     toplam += sayi;
7 }
▶ System.out.println("Toplam: " + toplam);
```

DEĞİŞKENLER

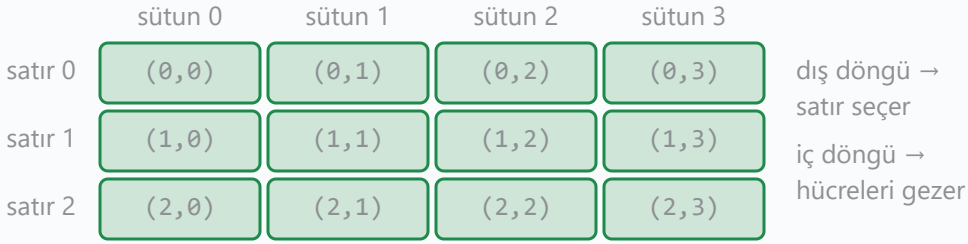
`toplam = 8`

ÇIKTI

```
Toplam: 8
```

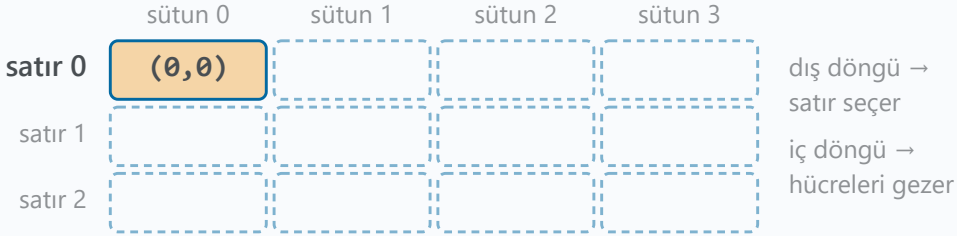
İç İçe Döngü

```
for (int satir = 0; satir < 3; satir++) {
  for (int sutun = 0; sutun < 4; sutun++) {
    System.out.printf("(%d,%d) ", satir, sutun);
  }
  System.out.println();
}
```



- İç döngü, dış döngünün **her** turunda baştan sona tamamlanır.
- Gövde satır × sutun = 3 × 4 = 12 kez çalışır.

Kod İzleme: 3 × 4 ızgara



Dış döngü **satir = 0**; iç döngü sutun = 0'dan başlar → (0,0).

Adım 1 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶     System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

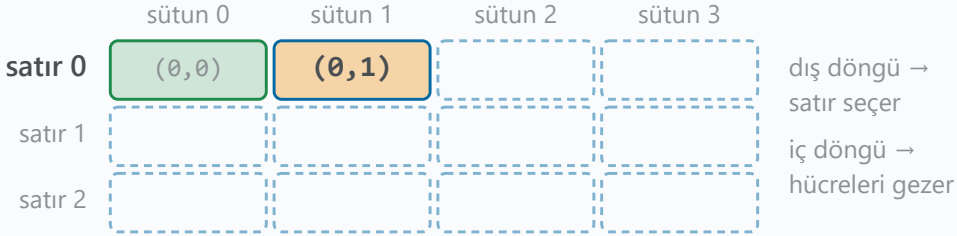
DEĞİŞKENLER

`satir = 0` `sutun = 0` `printf sayisi = 1`

ÇIKTI

```
(0,0)
```

Kod İzleme: 3 × 4 ızgara



sutun++ → 1 < 4 → true → **(0,1)** yazılır.

Adım 2 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶     System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

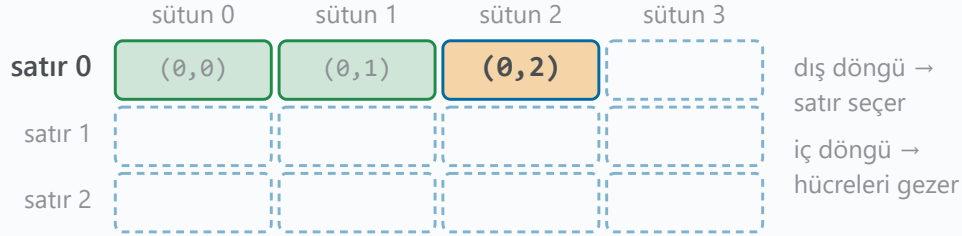
DEĞİŞKENLER

`satir = 0` `sutun = 1` `printf sayisi = 2`

ÇIKTI

```
(0,0) (0,1)
```

Kod İzleme: 3 × 4 ızgara



sutun++ → 2 < 4 → true → **(0,2)** yazılır.

Adım 3 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶       System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

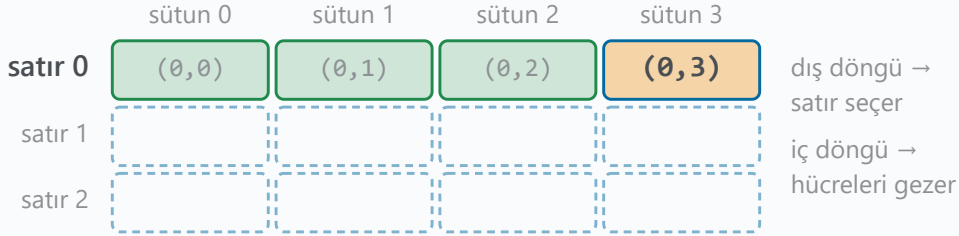
DEĞİŞKENLER

`satir = 0` `sutun = 2` `printf sayisi = 3`

ÇIKTI

```
(0,0) (0,1) (0,2)
```

Kod İzleme: 3 × 4 ızgara



sutun++ → 3 < 4 → true → **(0,3)** yazılır.

Adım 4 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶     System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

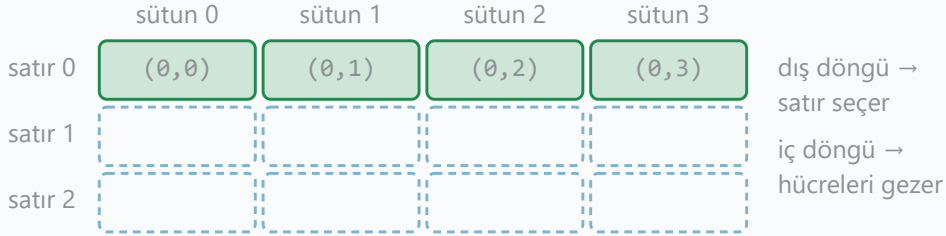
DEĞİŞKENLER

`satir = 0` `sutun = 3` `printf sayisi = 4`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
```

Kod İzleme: 3 × 4 ızgara



sutun = 4 < 4 → false: iç döngü bitti. `println` satırı sonlandırır.

Adım 5 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
3     System.out.printf("(%d,%d) ", satir, sutun);
4   }
▶   System.out.println();
6 }
```

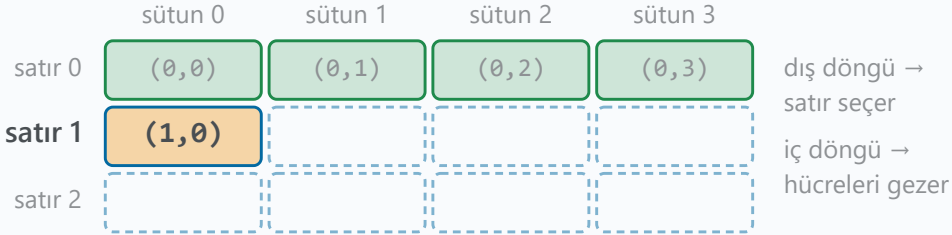
DEĞİŞKENLER

`satir = 0` `printf sayisi = 4`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
```

Kod İzleme: 3 × 4 ızgara



satir++ → **satir = 1**; iç döngü **yeniden** sutun = 0'dan başlar.

Adım 6 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶     System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

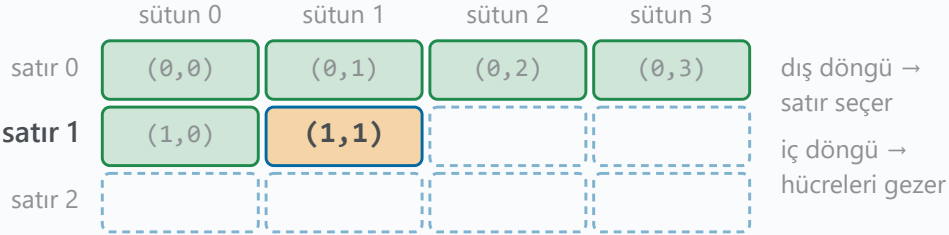
DEĞİŞKENLER

`satir = 1` `sutun = 0` `printf sayisi = 5`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0)
```

Kod İzleme: 3 × 4 ızgara



sutun++ → 1 < 4 → true → **(1,1)** yazılır.

Adım 7 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶       System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

DEĞİŞKENLER

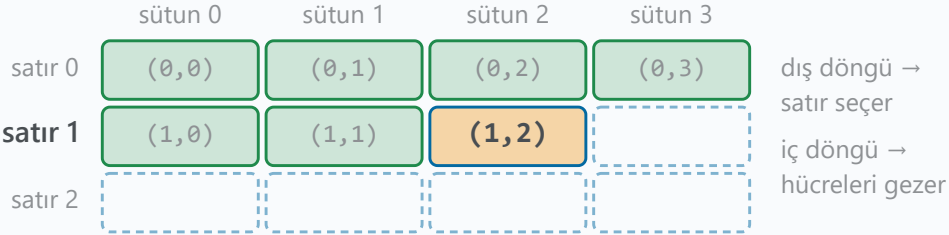
`satir = 1` `sutun = 1`

`printf sayisi = 6`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1)
```

Kod İzleme: 3 × 4 ızgara



sutun++ → 2 < 4 → true → **(1,2)** yazılır.

Adım 8 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶     System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

DEĞİŞKENLER

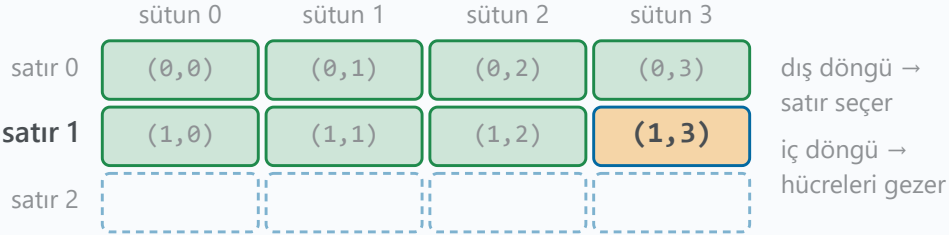
`satir = 1` `sutun = 2`

`printf sayisi = 7`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1) (1,2)
```

Kod İzleme: 3 × 4 ızgara



sutun++ → 3 < 4 → true → **(1,3)** yazılır.

Adım 9 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶       System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

DEĞİŞKENLER

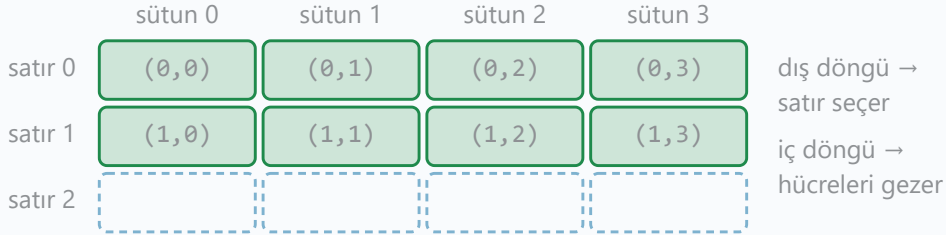
`satir = 1` `sutun = 3`

`printf sayisi = 8`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1) (1,2) (1,3)
```

Kod İzleme: 3 × 4 ızgara



sutun = 4 < 4 → false: iç döngü bitti. `println` satırı sonlandırır.

Adım 10 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
3     System.out.printf("(%d,%d) ", satir, sutun);
4   }
▶   System.out.println();
6 }
```

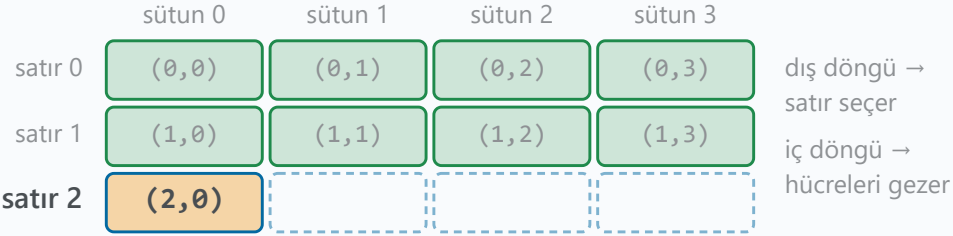
DEĞİŞKENLER

`satir = 1` `printf sayisi = 8`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1) (1,2) (1,3)
```

Kod İzleme: 3 × 4 ızgara



satir++ → **satir = 2**; iç döngü **yeniden** sutun = 0'dan başlar.

Adım 11 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶     System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

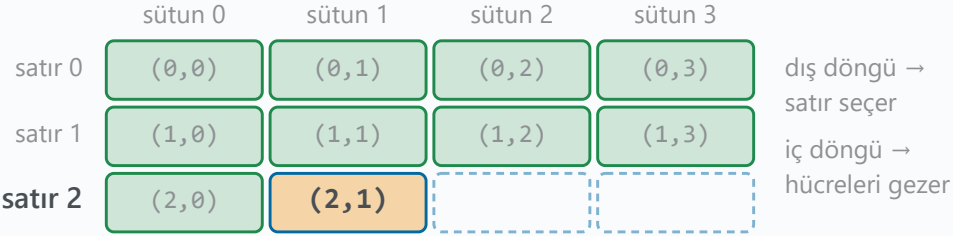
DEĞİŞKENLER

`satir = 2` `sutun = 0` `printf sayisi = 9`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1) (1,2) (1,3)
(2,0)
```

Kod İzleme: 3 × 4 ızgara



sutun++ → 1 < 4 → true → **(2,1)** yazılır.

Adım 12 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶       System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

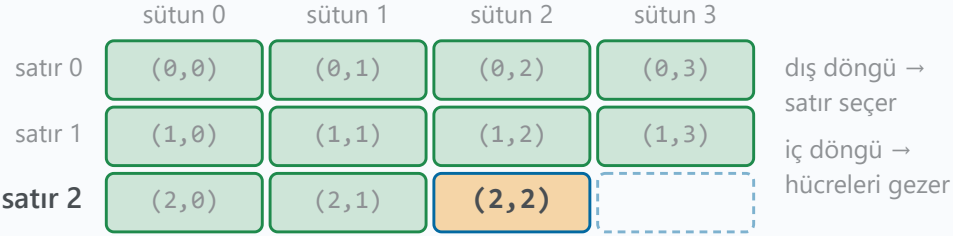
DEĞİŞKENLER

`satir = 2` `sutun = 1` `printf sayisi = 10`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1) (1,2) (1,3)
(2,0) (2,1)
```

Kod İzleme: 3 × 4 ızgara



sutun++ → 2 < 4 → true → **(2,2)** yazılır.

Adım 13 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶     System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

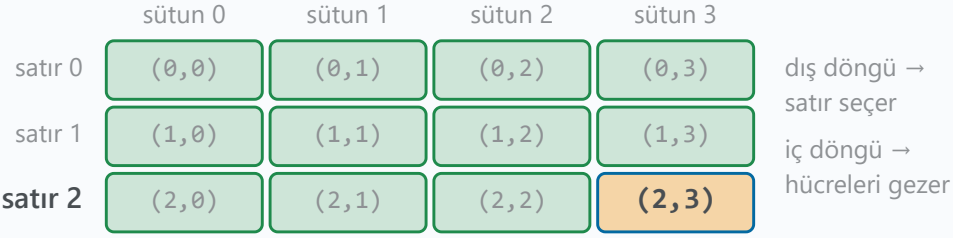
DEĞİŞKENLER

`satir = 2` `sutun = 2` `printf sayisi = 11`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1) (1,2) (1,3)
(2,0) (2,1) (2,2)
```

Kod İzleme: 3 × 4 ızgara



sutun++ → 3 < 4 → true → **(2,3)** yazılır.

Adım 14 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
▶       System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

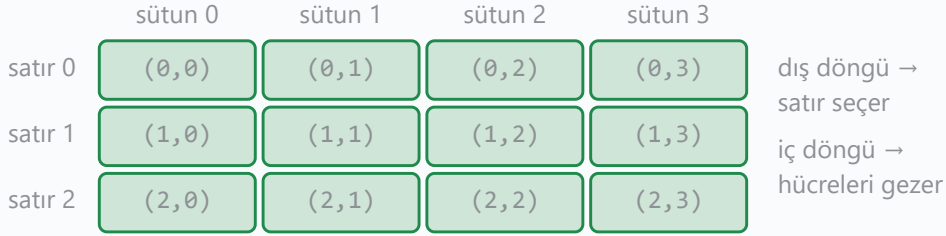
DEĞİŞKENLER

`satir = 2` `sutun = 3` `printf sayisi = 12`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1) (1,2) (1,3)
(2,0) (2,1) (2,2) (2,3)
```

Kod İzleme: 3 × 4 ızgara



sutun = 4 < 4 → false: iç döngü bitti. `println` satırı sonlandırır.

Adım 15 / 16

```
1 for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
3     System.out.printf("(%d,%d) ", satir, sutun);
4   }
▶   System.out.println();
6 }
```

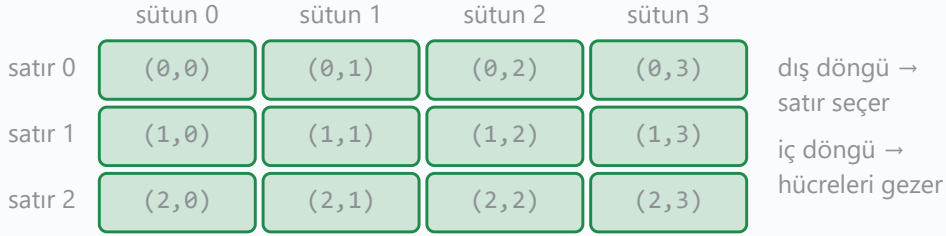
DEĞİŞKENLER

`satir = 2` `printf sayisi = 12`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1) (1,2) (1,3)
(2,0) (2,1) (2,2) (2,3)
```

Kod İzleme: 3 × 4 ızgara



satir = 3 < 3 → false: dış döngü de bitti. Toplam 3 × 4 = **12** printf.

Adım 16 / 16

```
▶ for (int satir = 0; satir < 3; satir++) {
2   for (int sutun = 0; sutun < 4; sutun++) {
3     System.out.printf("(%d,%d) ", satir, sutun);
4   }
5   System.out.println();
6 }
```

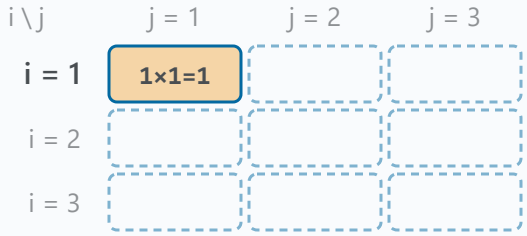
DEĞİŞKENLER

`printf sayisi = 12`

ÇIKTI

```
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1) (1,2) (1,3)
(2,0) (2,1) (2,2) (2,3)
```

Çarpım Tablosu (iç içe döngü, 3 × 3)



hücre (i, j) = i × j
%4d: 4 karakter genişlik

Dış döngü **i = 1**; iç döngü j = 1'den başlar: 1 × 1 = 1.

Adım 1 / 13

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
▶         System.out.printf("%4d", i * j);
4     }
5     System.out.println();
6 }
```

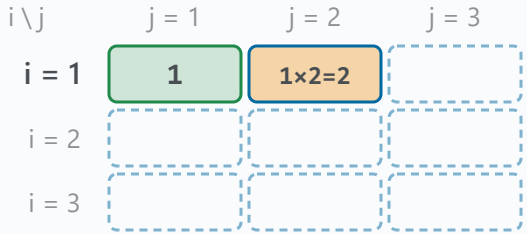
DEĞİŞKENLER

`i = 1` `j = 1`

ÇIKTI

```
   1
```

Çarpım Tablosu (iç içe döngü, 3 × 3)



hücre (i, j) = i × j
%4d: 4 karakter genişlik

j = 2: 1 × 2 = **2**, 4 karakterlik alana sağa yaslı yazılır.

Adım 2 / 13

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
▶         System.out.printf("%4d", i * j);
4     }
5     System.out.println();
6 }
```

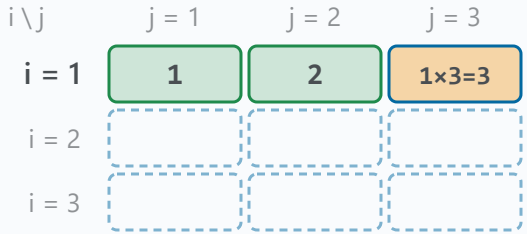
DEĞİŞKENLER

`i = 1` `j = 2`

ÇIKTI

```
   1   2
```

Çarpım Tablosu (iç içe döngü, 3 × 3)



hücre (i, j) = i × j
%4d: 4 karakter genişlik

j = 3: 1 × 3 = **3**, 4 karakterlik alana sağa yaslı yazılır.

Adım 3 / 13

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
▶         System.out.printf("%4d", i * j);
4     }
5     System.out.println();
6 }
```

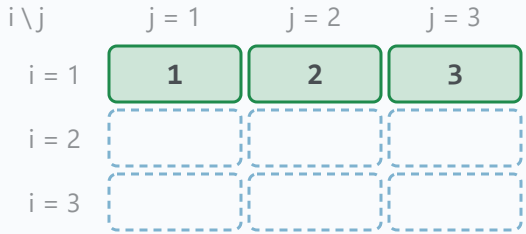
DEĞİŞKENLER

`i = 1` `j = 3`

ÇIKTI

```
   1   2   3
```

Çarpım Tablosu (iç içe döngü, 3 × 3)



hücre (i, j) = i × j
%4d: 4 karakter genişlik

j = 4 <= 3 → false; println ile satır biter.

Adım 4 / 13

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
3         System.out.printf("%4d", i * j);
4     }
▶     System.out.println();
6 }
```

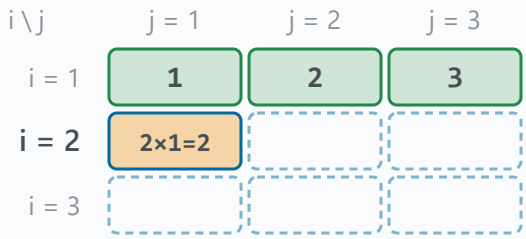
DEĞİŞKENLER

`i = 1`

ÇIKTI

```
   1   2   3
```

Çarpım Tablosu (iç içe döngü, 3 × 3)



hücre (i, j) = i × j
%4d: 4 karakter genişlik

Dış döngü **i = 2**; iç döngü j = 1'den başlar: 2 × 1 = 2.

Adım 5 / 13

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
▶         System.out.printf("%4d", i * j);
4     }
5     System.out.println();
6 }
```

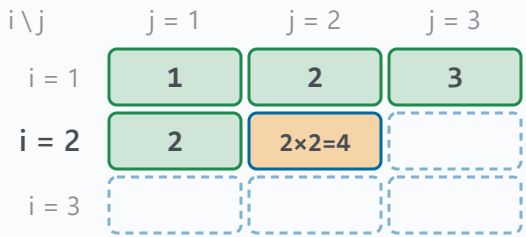
DEĞİŞKENLER

`i = 2` `j = 1`

ÇIKTI

```
   1   2   3
   2
```

Çarpım Tablosu (iç içe döngü, 3 × 3)



hücre (i, j) = i × j
%4d: 4 karakter genişlik

j = 2: 2 × 2 = **4**, 4 karakterlik alana sağa yaslı yazılır.

Adım 6 / 13

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
▶         System.out.printf("%4d", i * j);
4     }
5     System.out.println();
6 }
```

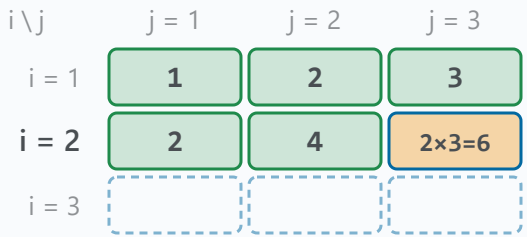
DEĞİŞKENLER

`i = 2` `j = 2`

ÇIKTI

```
   1   2   3
   2   4
```

Çarpım Tablosu (iç içe döngü, 3 × 3)



hücre (i, j) = i × j
%4d: 4 karakter genişlik

j = 3: 2 × 3 = **6**, 4 karakterlik alana sağa yaslı yazılır.

Adım 7 / 13

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
▶         System.out.printf("%4d", i * j);
4     }
5     System.out.println();
6 }
```

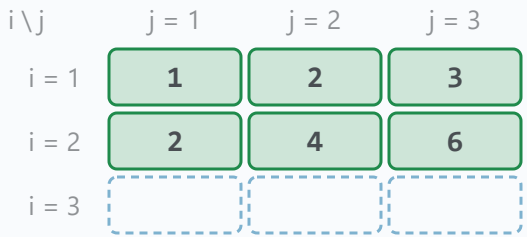
DEĞİŞKENLER

`i = 2` `j = 3`

ÇIKTI

```
   1   2   3
   2   4   6
```

Çarpım Tablosu (iç içe döngü, 3 × 3)



hücre (i, j) = i × j
%4d: 4 karakter genişlik

j = 4 <= 3 → false; println ile satır biter.


```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
3         System.out.printf("%4d", i * j);
4     }
▶     System.out.println();
6 }
```

DEĞİŞKENLER

`i = 2`

ÇIKTI

```
   1   2   3
   2   4   6
```

Çarpım Tablosu (iç içe döngü, 3 × 3)

i \ j	j = 1	j = 2	j = 3
i = 1	1	2	3
i = 2	2	4	6
i = 3	3×1=3		

hücre (i, j) = i × j
%4d: 4 karakter genişlik

Dış döngü **i = 3**; iç döngü j = 1'den başlar: 3 × 1 = 3.

Adım 9 / 13

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
          System.out.printf("%4d", i * j);
4     }
5     System.out.println();
6 }
```

DEĞİŞKENLER

`i = 3` `j = 1`

ÇIKTI

```
   1   2   3
   2   4   6
   3
```

Çarpım Tablosu (iç içe döngü, 3 × 3)

i \ j	j = 1	j = 2	j = 3
i = 1	1	2	3
i = 2	2	4	6
i = 3	3	3×2=6	

hücre (i, j) = i × j
%4d: 4 karakter genişlik

j = 2: 3 × 2 = **6**, 4 karakterlik alana sağa yaslı yazılır.

Adım 10 / 13

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
          System.out.printf("%4d", i * j);
4     }
5     System.out.println();
6 }
```

DEĞİŞKENLER

`i = 3` `j = 2`

ÇIKTI

```
   1   2   3
   2   4   6
   3   6
```

Çarpım Tablosu (iç içe döngü, 3 × 3)

i \ j	j = 1	j = 2	j = 3
i = 1	1	2	3
i = 2	2	4	6
i = 3	3	6	3×3=9

hücre (i, j) = i × j
%4d: 4 karakter genişlik

j = 3: 3 × 3 = **9**, 4 karakterlik alana sağa yaslı yazılır.


```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
▶         System.out.printf("%4d", i * j);
4     }
5     System.out.println();
6 }
```

DEĞİŞKENLER

`i = 3` `j = 3`

ÇIKTI

```
   1   2   3
   2   4   6
   3   6   9
```

Çarpım Tablosu (iç içe döngü, 3 × 3)

i \ j	j = 1	j = 2	j = 3
i = 1	1	2	3
i = 2	2	4	6
i = 3	3	6	9

hücre (i, j) = i × j
%4d: 4 karakter genişlik

j = 4 <= 3 → false; println ile satır biter.

Adım 12 / 13

```
1 for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
3         System.out.printf("%4d", i * j);
4     }
▶     System.out.println();
6 }
```

DEĞİŞKENLER

`i = 3`

ÇIKTI

```
   1   2   3
   2   4   6
   3   6   9
```

Çarpım Tablosu (iç içe döngü, 3 × 3)

i \ j	j = 1	j = 2	j = 3
i = 1	1	2	3
i = 2	2	4	6
i = 3	3	6	9

hücre (i, j) = i × j
%4d: 4 karakter genişlik

i = 4 <= 3 → false. Sınırları 10 yapınca klasik 10 × 10 çarpım tablosu çıkar.

Adım 13 / 13

```
▶ for (int i = 1; i <= 3; i++) {
2     for (int j = 1; j <= 3; j++) {
3         System.out.printf("%4d", i * j);
4     }
5     System.out.println();
6 }
```

DEĞİŞKENLER

—

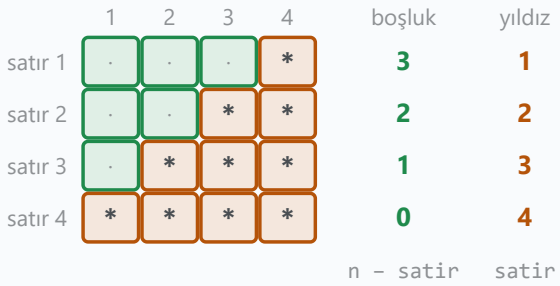
ÇIKTI

```
   1   2   3
   2   4   6
   3   6   9
```

ÖRNEK

Desen Problemini Çözme

1. Beklenen çıktıyı küçük bir n (ör. 4) için kareli kâğıda çizin.
2. Her satırda kaç **boşluk** ve kaç **karakter** olduğunu tabloya dökün.
3. Sayıları satır numarasıyla ilişkilendirin (3, 2, 1, 0 → n − satır).
4. Dış döngü = satır; her parça için bir iç döngü.

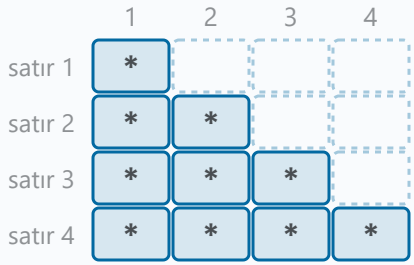


ÖRNEK

Artan Üçgen

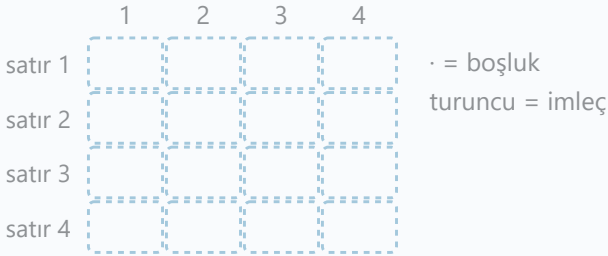
```
int n = 4;
for (int satir = 1; satir <= n; satir++) {
    for (int sutun = 1; sutun <= satir; sutun++) {
        System.out.print("*");
    }
    System.out.println();
}
```

- Satır r ise r yıldız yazılır.
- İç döngünün üst sınırı dış döngü değişkenidir.



ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: artan üçgen (n = 4)



n = 4 satırlık üçgen. Izgarada her hücre bir karakter.

Adım 1 / 16

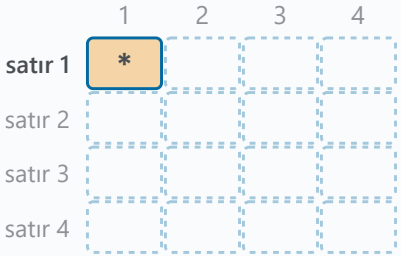
```
▶ int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
4     System.out.print("*");
5   }
6   System.out.println();
7 }
```

DEĞİŞKENLER

n = 4

ÇIKTI

Kod İzleme: artan üçgen (n = 4)



satir = 1: iç döngü sutun = 1 … 1 → bu satıra 1 yıldız.

Adım 2 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
▶     System.out.print("*");
5   }
6   System.out.println();
7 }
```

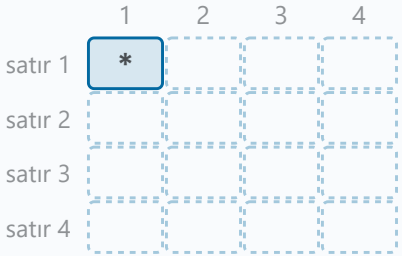
DEĞİŞKENLER

`n = 4` `satir = 1` `sutun = 1`

ÇIKTI

```
*
```

Kod İzleme: artan üçgen (n = 4)



sutun = 2 <= 1 → false. println ile alt satıra geçilir.

Adım 3 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
4     System.out.print("*");
5   }
▶   System.out.println();
7 }
```

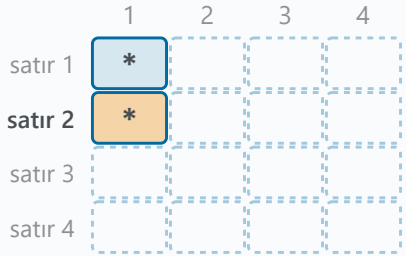
DEĞİŞKENLER

`n = 4` `satir = 1`

ÇIKTI

```
*
```

Kod İzleme: artan üçgen (n = 4)



satir = 2: iç döngü sutun = 1 … 2 → bu satıra 2 yıldız.

Adım 4 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
▶     System.out.print("*");
5   }
6   System.out.println();
7 }
```

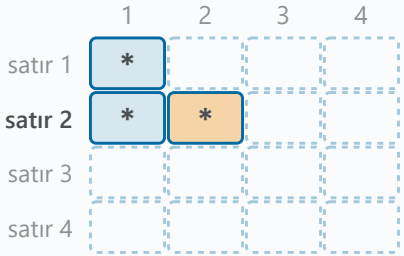
DEĞİŞKENLER

`n = 4` `satir = 2` `sutun = 1`

ÇIKTI

```
*
*
```

Kod İzleme: artan üçgen (n = 4)



sutun = 2 <= 2 → true → *

Adım 5 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
▶     System.out.print("*");
5   }
6   System.out.println();
7 }
```

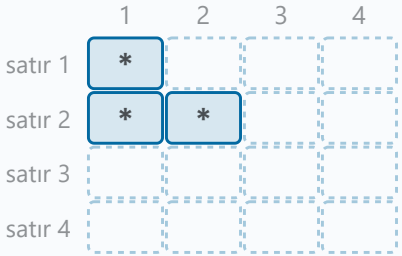
DEĞİŞKENLER

`n = 4` `satir = 2` `sutun = 2`

ÇIKTI

```
*
**
```

Kod İzleme: artan üçgen (n = 4)



sutun = 3 <= 2 → false. println ile alt satıra geçilir.

Adım 6 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
4     System.out.print("*");
5   }
▶   System.out.println();
7 }
```

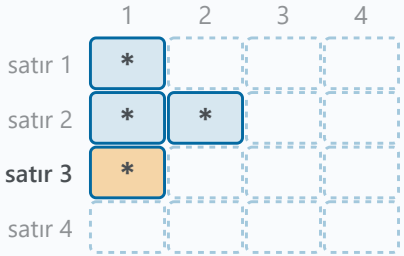
DEĞİŞKENLER

`n = 4` `satir = 2`

ÇIKTI

```
*
**
```

Kod İzleme: artan üçgen (n = 4)



satir = 3: iç döngü sutun = 1 … 3 → bu satıra 3 yıldız.

Adım 7 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
▶       System.out.print("*");
5   }
6   System.out.println();
7 }
```

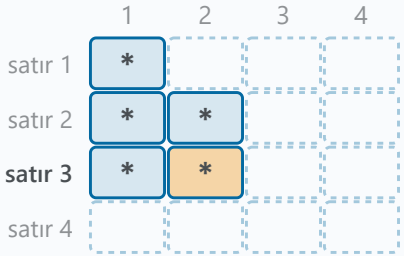
DEĞİŞKENLER

n = 4 | satir = 3 | sutun = 1

ÇIKTI

```
*
**
*
```

Kod İzleme: artan üçgen (n = 4)



sutun = 2 <= 3 → true → *

Adım 8 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
▶     System.out.print("*");
5   }
6   System.out.println();
7 }
```

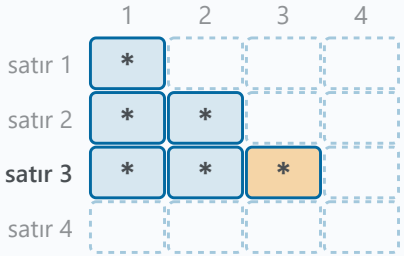
DEĞİŞKENLER

`n = 4` `satir = 3` `sutun = 2`

ÇIKTI

```
*
**
**
```

Kod İzleme: artan üçgen (n = 4)



sutun = 3 <= 3 → true → *

Adım 9 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
▶     System.out.print("*");
5   }
6   System.out.println();
7 }
```

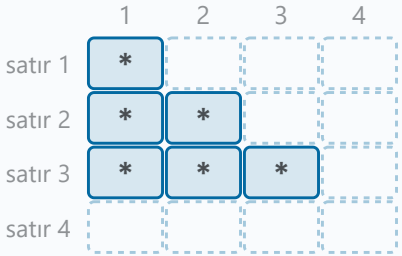
DEĞİŞKENLER

`n = 4` `satir = 3` `sutun = 3`

ÇIKTI

```
*
**
***
```

Kod İzleme: artan üçgen (n = 4)



sutun = 4 <= 3 → false. println ile alt satıra geçilir.

Adım 10 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
4     System.out.print("*");
5   }
▶   System.out.println();
7 }
```

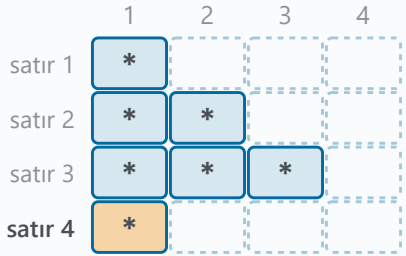
DEĞİŞKENLER

`n = 4` `satir = 3`

ÇIKTI

```
*
**
***
```

Kod İzleme: artan üçgen (n = 4)



satir = 4: iç döngü sutun = 1 … 4 → bu satıra 4 yıldız.

Adım 11 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
▶     System.out.print("*");
5   }
6   System.out.println();
7 }
```

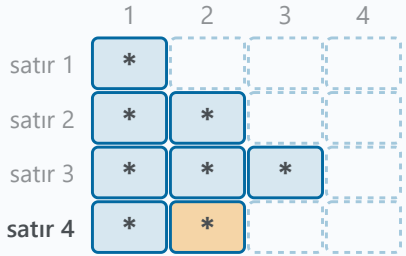
DEĞİŞKENLER

`n = 4` `satir = 4` `sutun = 1`

ÇIKTI

```
*
**
***
*
```

Kod İzleme: artan üçgen (n = 4)



sutun = 2 <= 4 → true → *

Adım 12 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
▶     System.out.print("*");
5   }
6   System.out.println();
7 }
```

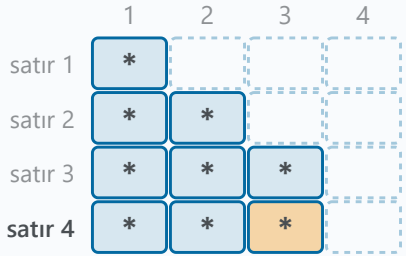
DEĞİŞKENLER

`n = 4` `satir = 4` `sutun = 2`

ÇIKTI

```
*
**
***
**
```

Kod İzleme: artan üçgen (n = 4)



sutun = 3 <= 4 → true → *

Adım 13 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
▶     System.out.print("*");
5   }
6   System.out.println();
7 }
```

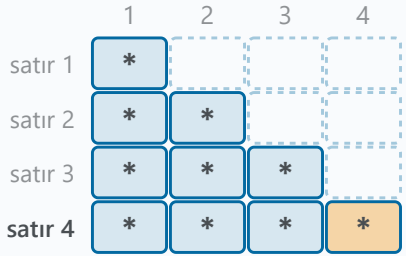
DEĞİŞKENLER

`n = 4` `satir = 4` `sutun = 3`

ÇIKTI

```
*
**
***
***
```

Kod İzleme: artan üçgen (n = 4)



sutun = 4 <= 4 → true → *

Adım 14 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
▶     System.out.print("*");
5   }
6   System.out.println();
7 }
```

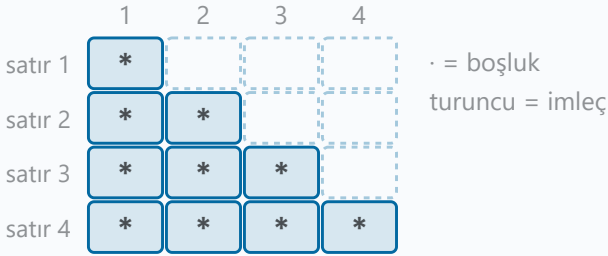
DEĞİŞKENLER

`n = 4` `satir = 4` `sutun = 4`

ÇIKTI

```
*
**
***
****
```

Kod İzleme: artan üçgen (n = 4)



sutun = 5 <= 4 → false. println ile alt satıra geçilir.

Adım 15 / 16

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
4     System.out.print("*");
5   }
▶   System.out.println();
7 }
```

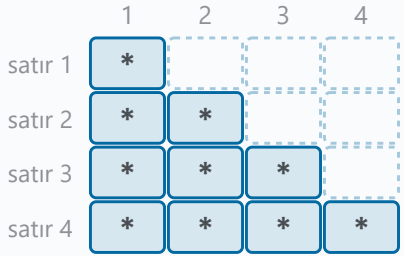
DEĞİŞKENLER

`n = 4` `satir = 4`

ÇIKTI

```
*
**
***
****
```

Kod İzleme: artan üçgen (n = 4)



satir = 5 <= 4 → false: bitti. Toplam 1 + 2 + 3 + 4 = 10 yıldız.

Adım 16 / 16

```
1 int n = 4;
▶ for (int satir = 1; satir <= n; satir++) {
3   for (int sutun = 1; sutun <= satir; sutun++) {
4     System.out.print("*");
5   }
6   System.out.println();
7 }
```

DEĞİŞKENLER

n = 4

ÇIKTI

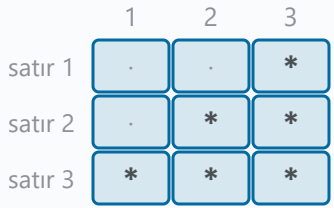
```
*
**
***
****
```

ÖRNEK

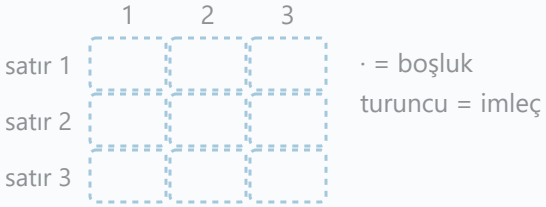
Sağa Yaslı Üçgen

```
int n = 3;
for (int satir = 1; satir <= n; satir++) {
    for (int bosluk = 0; bosluk < n - satir; bosluk++)
        System.out.print(" ");
    for (int yildiz = 0; yildiz < satir; yildiz++)
        System.out.print("*");
    System.out.println();
}
```

- Her satır boşluk bölümü ve karakter bölümünden oluşur.
- Boşluk azalırken yıldız artar.



Kod İzleme: sağa yaslı üçgen (n = 3)



n = 3. Her satır: önce `n - satir` boşluk, sonra `satir` yıldız.

Adım 1 / 14

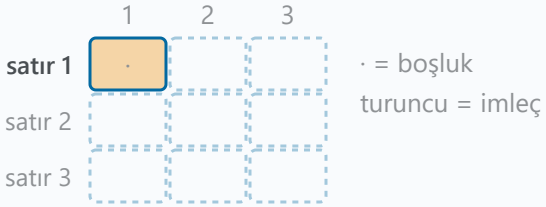
```
▶ int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
6     System.out.print("*");
7   System.out.println();
8 }
```

DEĞİŞKENLER

n = 3

ÇIKTI

Kod İzleme: sağa yaslı üçgen (n = 3)



satir = 1: 2 boşluk gerekiyor. bosluk = 0 → ' ' yazılır.

Adım 2 / 14

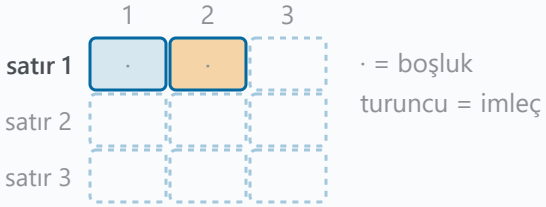
```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
    ▶   System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
6     System.out.print("*");
7   System.out.println();
8 }
```

DEĞİŞKENLER

`n = 3` `satir = 1` `bosluk = 0`

ÇIKTI

Kod İzleme: sağa yaslı üçgen (n = 3)



bosluk = 1 < 2 → ' ' yazılır.

Adım 3 / 14

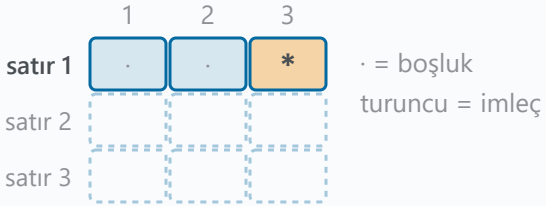
```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
    ▶   System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
6     System.out.print("*");
7   System.out.println();
8 }
```

DEĞİŞKENLER

`n = 3` `satir = 1` `bosluk = 1`

ÇIKTI

Kod İzleme: sağa yaslı üçgen (n = 3)



Boşluklar bitti; 1 yıldız: yildiz = 0 → *

Adım 4 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
▶     System.out.print("*");
7   System.out.println();
8 }
```

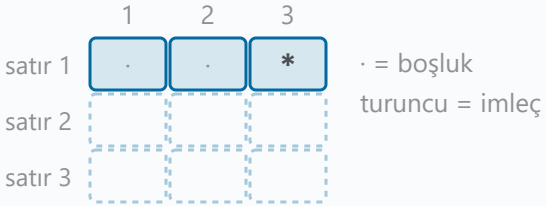
DEĞİŞKENLER

`n = 3` `satir = 1` `yildiz = 0`

ÇIKTI

```
  *
```

Kod İzleme: sağa yaslı üçgen (n = 3)



Satır tamam (her satır n = 3 karakter): println.

Adım 5 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
6     System.out.print("*");
▶   System.out.println();
8 }
```

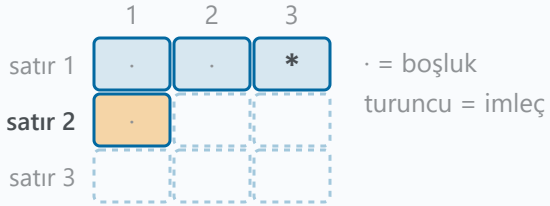
DEĞİŞKENLER

`n = 3` `satir = 1`

ÇIKTI

```
  *
```

Kod İzleme: sağa yaslı üçgen (n = 3)



satir = 2: 1 boşluk gerekiyor. bosluk = 0 → ' ' yazılır.

Adım 6 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
    ▶   System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
6     System.out.print("*");
7   System.out.println();
8 }
```

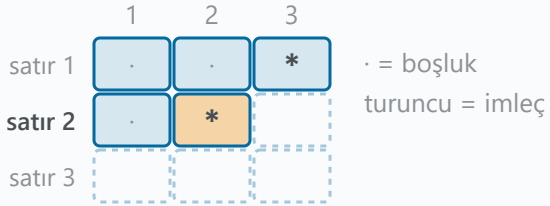
DEĞİŞKENLER

n = 3 | satir = 2 | bosluk = 0

ÇIKTI

```
  *
```

Kod İzleme: sağa yaslı üçgen (n = 3)



Boşluklar bitti; 2 yıldız: yildiz = 0 → *

Adım 7 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
▶     System.out.print("*");
7   System.out.println();
8 }
```

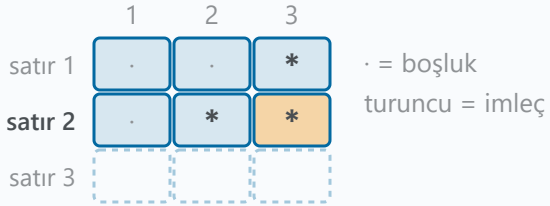
DEĞİŞKENLER

`n = 3` `satir = 2` `yildiz = 0`

ÇIKTI

```
  *
 *
```

Kod İzleme: sağa yaslı üçgen (n = 3)



yildiz = 1 < 2 → *

Adım 8 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
▶     System.out.print("*");
7   System.out.println();
8 }
```

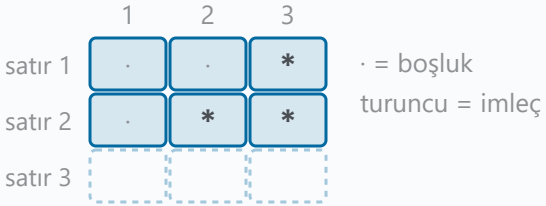
DEĞİŞKENLER

`n = 3` `satir = 2` `yildiz = 1`

ÇIKTI

```
  *
 **
```

Kod İzleme: sağa yaslı üçgen (n = 3)



Satır tamam (her satır n = 3 karakter): println.

Adım 9 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
6     System.out.print("*");
▶   System.out.println();
8 }
```

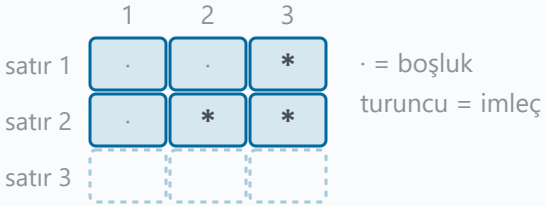
DEĞİŞKENLER

`n = 3` `satir = 2`

ÇIKTI

```
  *
 **
```

Kod İzleme: sağa yaslı üçgen (n = 3)



satir = 3: bosluk < 3 − 3 = 0 → false: bu satırda hiç boşluk yok.

Adım 10 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
▶   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
6     System.out.print("*");
7   System.out.println();
8 }
```

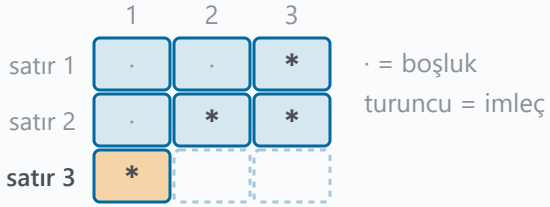
DEĞİŞKENLER

`n = 3` `satir = 3` `bosluk = 0`

ÇIKTI

```
  *
 **
```

Kod İzleme: sağa yaslı üçgen (n = 3)



Boşluklar bitti; 3 yıldız: yildiz = 0 → *

Adım 11 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
▶     System.out.print("*");
7   System.out.println();
8 }
```

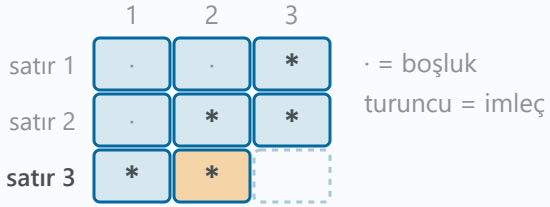
DEĞİŞKENLER

n = 3 | satir = 3 | yildiz = 0

ÇIKTI

```
  *
 **
*
```

Kod İzleme: sağa yaslı üçgen (n = 3)



yildiz = 1 < 3 → *

Adım 12 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
▶     System.out.print("*");
7   System.out.println();
8 }
```

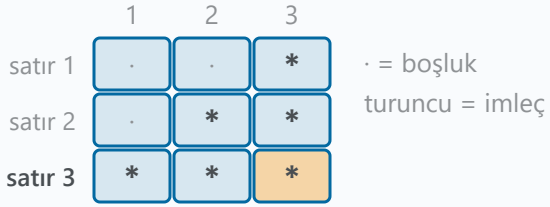
DEĞİŞKENLER

`n = 3` `satir = 3` `yildiz = 1`

ÇIKTI

```
  *
 **
**
```

Kod İzleme: sağa yaslı üçgen (n = 3)



yildiz = 2 < 3 → *

Adım 13 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
▶     System.out.print("*");
7   System.out.println();
8 }
```

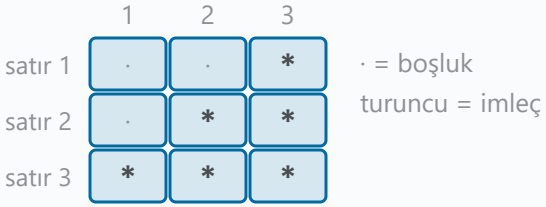
DEĞİŞKENLER

`n = 3` `satir = 3` `yildiz = 2`

ÇIKTI

```
  *
 **
***
```

Kod İzleme: sağa yaslı üçgen (n = 3)



Satır tamam (her satır n = 3 karakter): println.

Adım 14 / 14

```
1 int n = 3;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int bosluk = 0; bosluk < n - satir; bosluk++)
4     System.out.print(" ");
5   for (int yildiz = 0; yildiz < satir; yildiz++)
6     System.out.print("*");
▶   System.out.println();
8 }
```

DEĞİŞKENLER

`n = 3` `satir = 3`

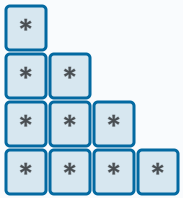
ÇIKTI

```
  *
 **
***
```

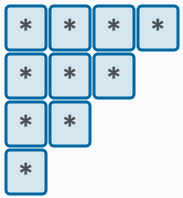
ÖRNEK

Satıra Göre Sayılar

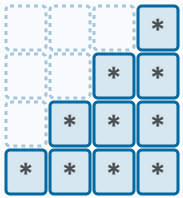
Desen	Boşluk	Karakter
Artan üçgen	0	satır
Azalan üçgen	0	n - satır + 1
Sağa yaslı	n - satır	satır
Piramit	n - satır	2 × satır - 1



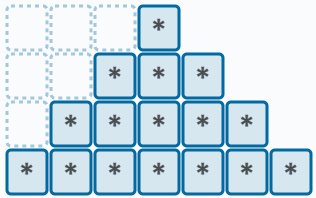
Artan üçgen



Azalan üçgen



Sağa yaslı



Piramit

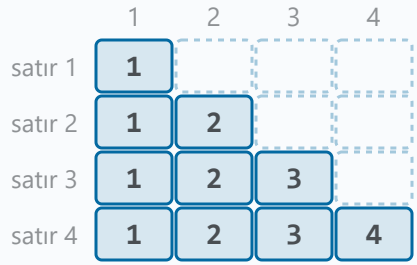
n = 4 · kesikli kare: boşluk, dolu kare: karakter

ÖRNEK

Sayı Üçgeni

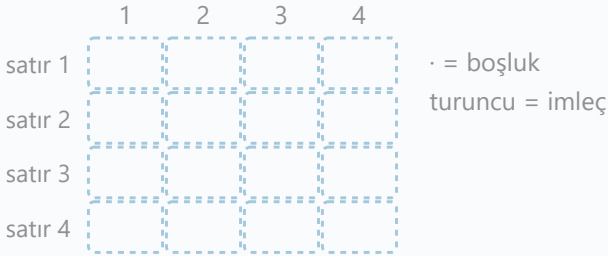
```
int n = 4;
for (int satir = 1; satir <= n; satir++) {
    for (int deger = 1; deger <= satir; deger++) {
        System.out.print(deger + " ");
    }
    System.out.println();
}
```

- Yazdırılan değer sütun değişkeninden gelir.
- Aynı yapı karakter yerine sayı üretir.



ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: sayı üçgeni (n = 4)



Her hücreye bir `deger + " "` (sayı ve boşluk) yazılır.

Adım 1 / 15

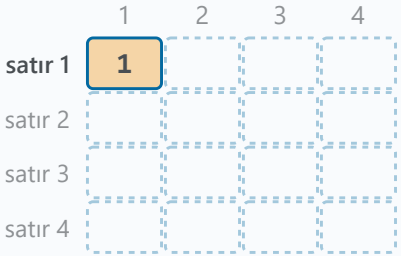
```
▶ int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
4     System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

DEĞİŞKENLER

n = 4

ÇIKTI

Kod İzleme: sayı üçgeni (n = 4)



· = boşluk
turuncu = imleç

satir = 1: değer 1'den 1'e kadar. deger = 1 → "1 "

Adım 2 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
▶     System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

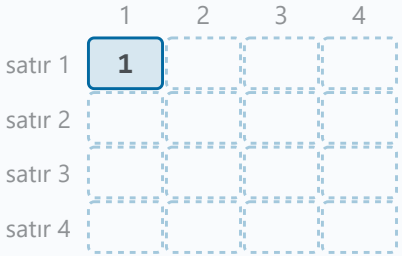
DEĞİŞKENLER

`n = 4` `satir = 1` `deger = 1`

ÇIKTI

```
1
```

Kod İzleme: sayı üçgeni (n = 4)



deger = 2 <= 1 → false; println.

Adım 3 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
4     System.out.print(deger + " ");
5   }
    System.out.println();
7 }
```

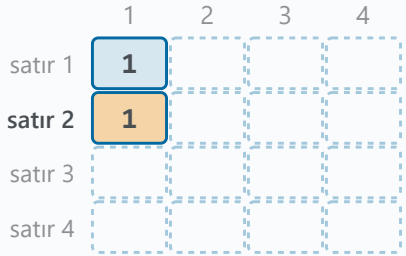
DEĞİŞKENLER

`n = 4` `satir = 1`

ÇIKTI

```
1
```

Kod İzleme: sayı üçgeni (n = 4)



satir = 2: değer 1'den 2'ye kadar. deger = 1 → "1 "

Adım 4 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
▶       System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

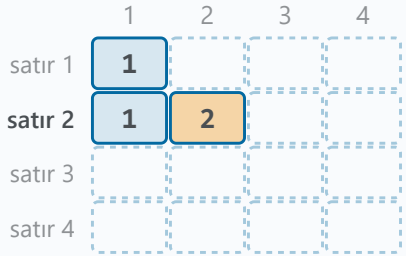
DEĞİŞKENLER

`n = 4` `satir = 2` `deger = 1`

ÇIKTI

```
1
1
```

Kod İzleme: sayı üçgeni (n = 4)



deger = 2 → “2 ” (yazılan sayı iç döngü değişkeni).

Adım 5 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
▶       System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

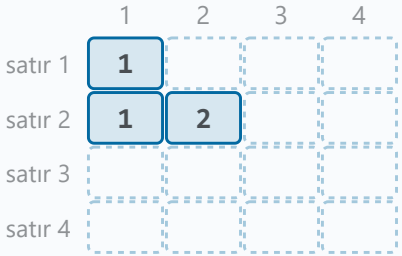
DEĞİŞKENLER

`n = 4` `satir = 2` `deger = 2`

ÇIKTI

```
1
1 2
```

Kod İzleme: sayı üçgeni (n = 4)



deger = 3 <= 2 → false; println.

Adım 6 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
4     System.out.print(deger + " ");
5   }
▶   System.out.println();
7 }
```

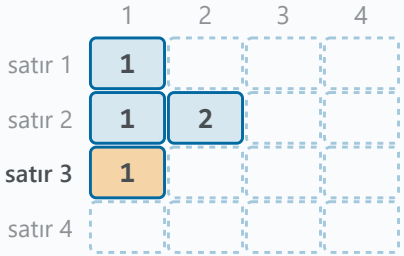
DEĞİŞKENLER

`n = 4` `satir = 2`

ÇIKTI

```
1
1 2
```

Kod İzleme: sayı üçgeni (n = 4)



satir = 3: değer 1'den 3'e kadar. deger = 1 → "1 "

Adım 7 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
▶       System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

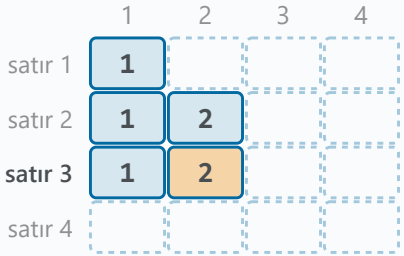
DEĞİŞKENLER

n = 4 | satir = 3 | deger = 1

ÇIKTI

```
1
1 2
1
```

Kod İzleme: sayı üçgeni (n = 4)



deger = 2 → “2 ” (yazılan sayı iç döngü değişkeni).

Adım 8 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
▶     System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

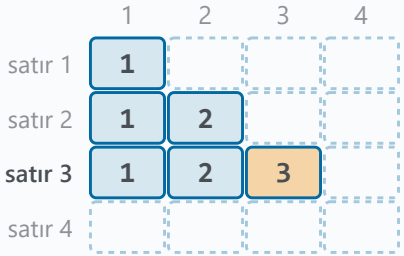
DEĞİŞKENLER

`n = 4` `satir = 3` `deger = 2`

ÇIKTI

```
1
1 2
1 2
```

Kod İzleme: sayı üçgeni (n = 4)



deger = 3 → “3 ” (yazılan sayı iç döngü değişkeni).

Adım 9 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
▶       System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

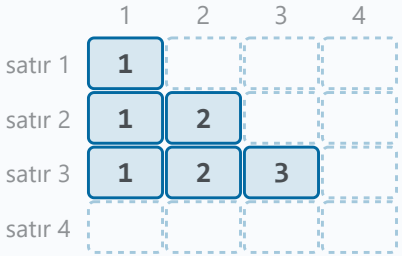
DEĞİŞKENLER

`n = 4` `satir = 3` `deger = 3`

ÇIKTI

```
1
1 2
1 2 3
```

Kod İzleme: sayı üçgeni (n = 4)



deger = 4 <= 3 → false; println.

Adım 10 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
4     System.out.print(deger + " ");
5   }
▶   System.out.println();
7 }
```

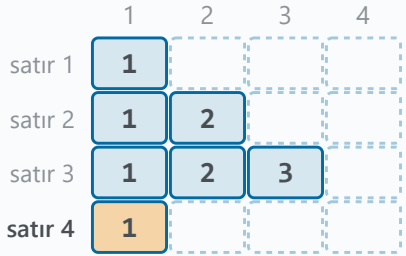
DEĞİŞKENLER

`n = 4` `satir = 3`

ÇIKTI

```
1
1 2
1 2 3
```

Kod İzleme: sayı üçgeni (n = 4)



satir = 4: değer 1'den 4'e kadar. deger = 1 → "1 "


```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
▶       System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

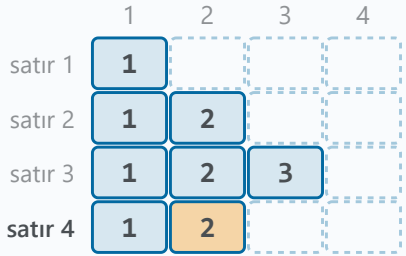
DEĞİŞKENLER

`n = 4` `satir = 4` `deger = 1`

ÇIKTI

```
1
1 2
1 2 3
1
```

Kod İzleme: sayı üçgeni (n = 4)



deger = 2 → “2 ” (yazılan sayı iç döngü değişkeni).

Adım 12 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
▶       System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

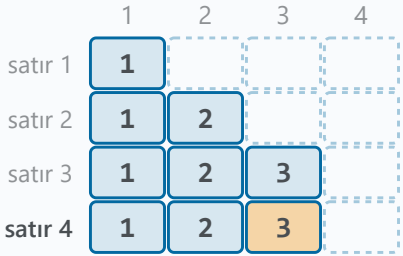
DEĞİŞKENLER

`n = 4` `satir = 4` `deger = 2`

ÇIKTI

```
1
1 2
1 2 3
1 2
```

Kod İzleme: sayı üçgeni (n = 4)



deger = 3 → “3 ” (yazılan sayı iç döngü değişkeni).

Adım 13 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
▶     System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

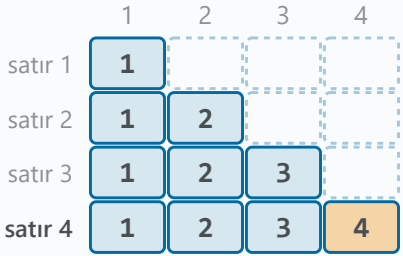
DEĞİŞKENLER

`n = 4` `satir = 4` `deger = 3`

ÇIKTI

```
1
1 2
1 2 3
1 2 3
```

Kod İzleme: sayı üçgeni (n = 4)



· = boşluk
turuncu = imleç

deger = 4 → “4 ” (yazılan sayı iç döngü değişkeni).

Adım 14 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
▶     System.out.print(deger + " ");
5   }
6   System.out.println();
7 }
```

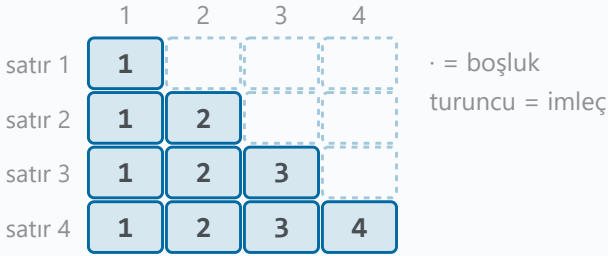
DEĞİŞKENLER

`n = 4` `satir = 4` `deger = 4`

ÇIKTI

```
1
1 2
1 2 3
1 2 3 4
```

Kod İzleme: sayı üçgeni (n = 4)



deger = 5 <= 4 → false; println.

Adım 15 / 15

```
1 int n = 4;
2 for (int satir = 1; satir <= n; satir++) {
3   for (int deger = 1; deger <= satir; deger++) {
4     System.out.print(deger + " ");
5   }
▶   System.out.println();
7 }
```

DEĞİŞKENLER

`n = 4` `satir = 4`

ÇIKTI

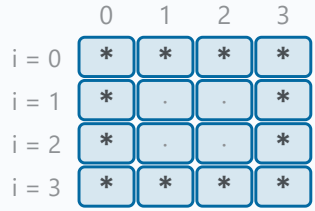
```
1
1 2
1 2 3
1 2 3 4
```

ÖRNEK

İçi Boş Kare

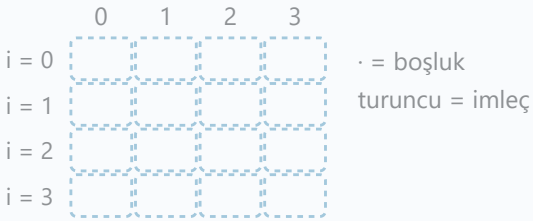
```
int n = 4;
for (int i = 0; i < n; i++) {
  for (int j = 0; j < n; j++) {
    if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
      System.out.print("*");
    else
      System.out.print(" ");
  }
  System.out.println();
}
```

- Yalnızca ilk/son satır veya ilk/son sütun yıldızdır.
- `||` : dört koşuldan biri true ise yeter.
- Diğer hücrelere boşluk yazılır; yoksa kare kayar.



ÖRNEKLER · KODU ADIM ADIM ÇALIŞTIR

Kod İzleme: içi boş kare (n = 4)



n = 4. Kenar: ilk/son satır (i = 0, 3) veya ilk/son sütun (j = 0, 3).

Adım 1 / 11

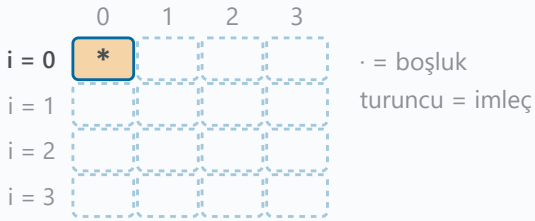
```
▶ int n = 4;
2 for (int i = 0; i < n; i++) {
3   for (int j = 0; j < n; j++) {
4     if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
5       System.out.print("*");
6     else
7       System.out.print(" ");
8   }
9   System.out.println();
10 }
```

DEĞİŞKENLER

n = 4

ÇIKTI

Kod İzleme: içi boş kare (n = 4)



i = 0, j = 0: **i == 0** → true → *

Adım 2 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

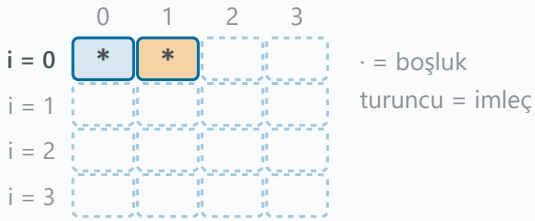
DEĞİŞKENLER

`n = 4` `i = 0` `j = 0`

ÇIKTI

```
*
```

Kod İzleme: içi boş kare (n = 4)



i = 0, j = 1: **i == 0** → true → *

Adım 3 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

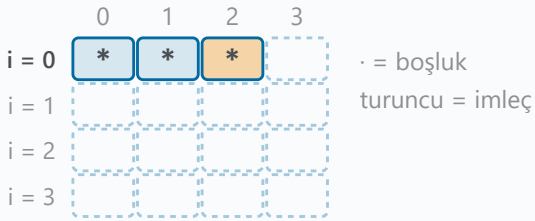
DEĞİŞKENLER

n = 4 | i = 0 | j = 1

ÇIKTI

```
**
```

Kod İzleme: içi boş kare (n = 4)



i = 0, j = 2: **i == 0** → true → *

Adım 4 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

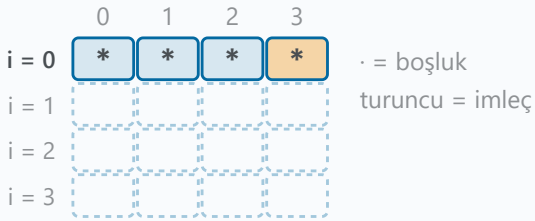
DEĞİŞKENLER

`n = 4` `i = 0` `j = 2`

ÇIKTI

```
***
```

Kod İzleme: içi boş kare (n = 4)



i = 0, j = 3: **i == 0** → true → *

Adım 5 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

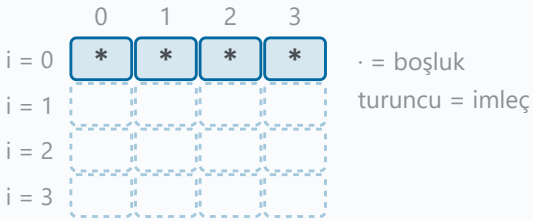
DEĞİŞKENLER

`n = 4` `i = 0` `j = 3`

ÇIKTI

```
****
```

Kod İzleme: içi boş kare (n = 4)



j = 4 < 4 → false; println.

Adım 6 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
4      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
5        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
▶    System.out.println();
10 }
```

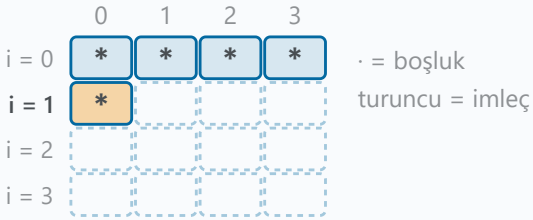
DEĞİŞKENLER

`n = 4` `i = 0`

ÇIKTI

```
****
```

Kod İzleme: içi boş kare (n = 4)



i = 1, j = 0: **j == 0** → true → *

Adım 7 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

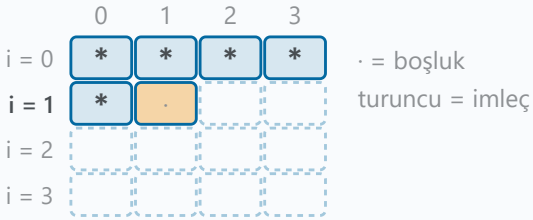
DEĞİŞKENLER

`n = 4` `i = 1` `j = 0`

ÇIKTI

```
****
*
```

Kod İzleme: içi boş kare (n = 4)



i = 1, j = 1: dört koşulun hepsi false → else → boşluk

Adım 8 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
5        System.out.print("*");
6      else
▶        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

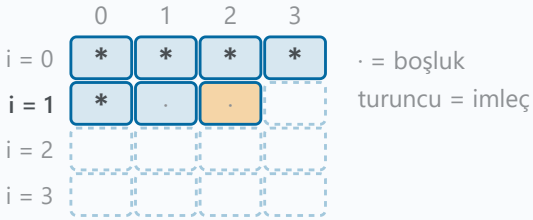
DEĞİŞKENLER

`n = 4` `i = 1` `j = 1`

ÇIKTI

```
****
*
```

Kod İzleme: içi boş kare (n = 4)



i = 1, j = 2: dört koşulun hepsi false → else → boşluk

Adım 9 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
5        System.out.print("*");
6      else
▶        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

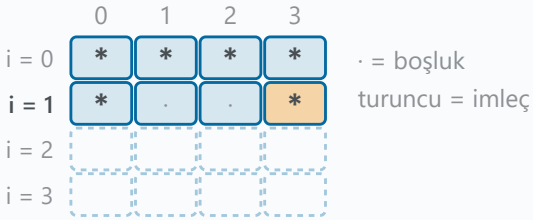
DEĞİŞKENLER

`n = 4` `i = 1` `j = 2`

ÇIKTI

```
****
*
```

Kod İzleme: içi boş kare (n = 4)



i = 1, j = 3: **j == 3** → true → *

Adım 10 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

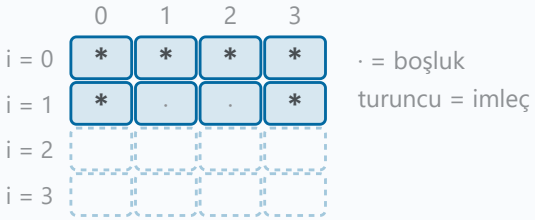
DEĞİŞKENLER

n = **4** | i = **1** | j = **3**

ÇIKTI

```
****
*  *
```

Kod İzleme: içi boş kare (n = 4)



j = 4 < 4 → false; println.

Adım 11 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
4      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
5        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
▶    System.out.println();
10 }
```

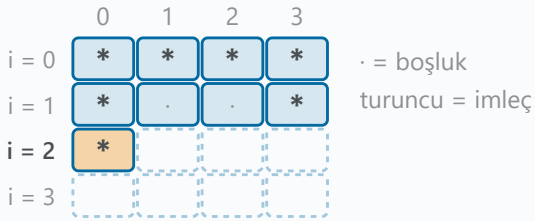
DEĞİŞKENLER

`n = 4` `i = 1`

ÇIKTI

```
****
*  *
```

Kod İzleme (devam): satır 2 ve 3



i = 2, j = 0: **j == 0** → true → *

Adım 1 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

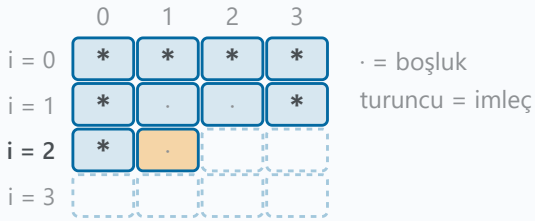
DEĞİŞKENLER

`n = 4` `i = 2` `j = 0`

ÇIKTI

```
****
*  *
*
```

Kod İzleme (devam): satır 2 ve 3



i = 2, j = 1: dört koşulun hepsi false → else → boşluk

Adım 2 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
5        System.out.print("*");
6      else
▶        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

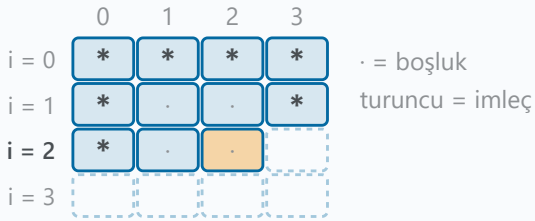
DEĞİŞKENLER

`n = 4` `i = 2` `j = 1`

ÇIKTI

```
****
*  *
*
```

Kod İzleme (devam): satır 2 ve 3



i = 2, j = 2: dört koşulun hepsi false → else → boşluk

Adım 3 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
5        System.out.print("*");
6      else
▶        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

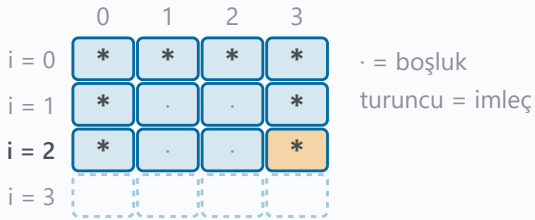
DEĞİŞKENLER

`n = 4` `i = 2` `j = 2`

ÇIKTI

```
****
*  *
*
```

Kod İzleme (devam): satır 2 ve 3



i = 2, j = 3: **j == 3** → true → *

Adım 4 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

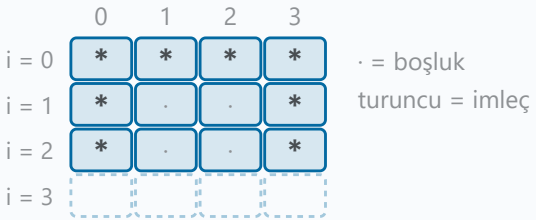
DEĞİŞKENLER

`n = 4` `i = 2` `j = 3`

ÇIKTI

```
****
*  *
*  *
```

Kod İzleme (devam): satır 2 ve 3



j = 4 < 4 → false; println.

Adım 5 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
4      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
5        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
▶    System.out.println();
10 }
```

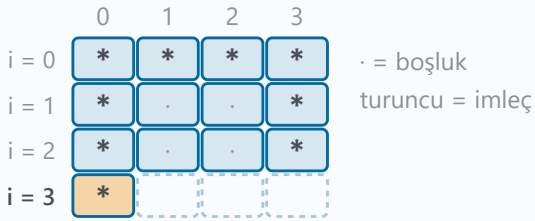
DEĞİŞKENLER

`n = 4` `i = 2`

ÇIKTI

```
****
*  *
*  *
```

Kod İzleme (devam): satır 2 ve 3



i = 3, j = 0: **i == 3** → true → *

Adım 6 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

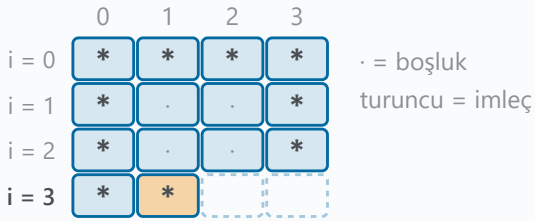
DEĞİŞKENLER

`n = 4` `i = 3` `j = 0`

ÇIKTI

```
****
*  *
*  *
*
```

Kod İzleme (devam): satır 2 ve 3



i = 3, j = 1: **i == 3** → true → *

Adım 7 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

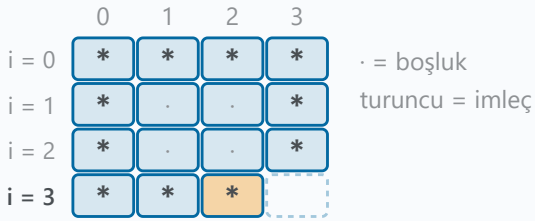
DEĞİŞKENLER

`n = 4` `i = 3` `j = 1`

ÇIKTI

```
****
*  *
*  *
**
```

Kod İzleme (devam): satır 2 ve 3



i = 3, j = 2: **i == 3** → true → *


```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

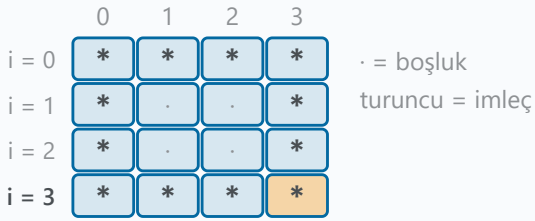
DEĞİŞKENLER

`n = 4` `i = 3` `j = 2`

ÇIKTI

```
****
*  *
*  *
***
```

Kod İzleme (devam): satır 2 ve 3



i = 3, j = 3: **i == 3** → true → *


```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
▶      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
▶        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

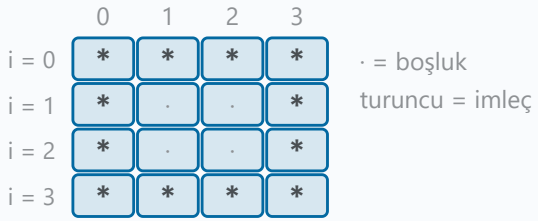
DEĞİŞKENLER

`n = 4` `i = 3` `j = 3`

ÇIKTI

```
****
*  *
*  *
****
```

Kod İzleme (devam): satır 2 ve 3



j = 4 < 4 → false; println.

Adım 10 / 11

```
1  int n = 4;
2  for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
4      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
5        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
▶    System.out.println();
10 }
```

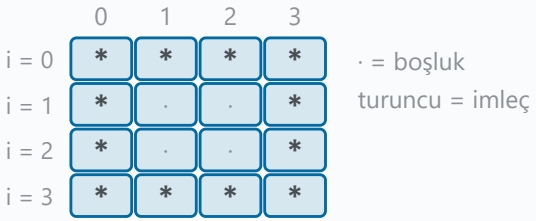
DEĞİŞKENLER

`n = 4` `i = 3`

ÇIKTI

```
****
*  *
*  *
****
```

Kod İzleme (devam): satır 2 ve 3



i = 4 < 4 → false: kare tamam. 16 hücrenin 12'si kenar.

Adım 11 / 11

```
1  int n = 4;
▶ for (int i = 0; i < n; i++) {
3    for (int j = 0; j < n; j++) {
4      if (i == 0 || i == n - 1 || j == 0 || j == n - 1)
5        System.out.print("*");
6      else
7        System.out.print(" ");
8    }
9    System.out.println();
10 }
```

DEĞİŞKENLER

`n = 4`

ÇIKTI

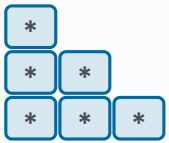
```
****
*  *
*  *
****
```

ÖRNEK

Sık Hatalar

- println çağrısını iç döngüye koymak her karakteri ayrı satıra yazar.
- < ve <= farkı bir eksik veya bir fazla karakter üretir.
- Boşluğu yıldızdan sonra yazmak hizalamayı tersine çevirir.
- Dış döngü değişkenini iç döngüde yanlış güncellemek akışı bozar.

doğru



println iç döngüde



ÖRNEK

Test Tasarımı

- Her problem için normal bir değer seçin.
- 0, 1, negatif ve üst sınıra yakın değerleri ayrı test edin.
- Beklenen sonucu kodu çalıştırmadan hesaplayın.
- Değişkenleri tur tur izleyerek ilk sapmayı bulun.

Basamak sayısı testi	sayı	beklenen
Normal	4096	4
Sıfır	0	1
Negatif	−507	3
Uç değer	−2147483648	10

ÖZET

Bölüm Özeti

- Her döngü ilk değer verme, koşul, gövde ve güncelleme parçalarından oluşur.
- Tekrar sayısı belliyse for, koşul sürdükçe while, en az bir tur gerekiyorsa do-while seçilir.
- Sayaç ve biriktirici kalıpları basamak, üs ve seri problemlerinde tekrar kullanılır.
- İç içe döngüde dış döngü satırı, iç döngü satırdaki öğeleri üretir.
- Rakam kalıbı (`% 10` , `/ 10`): basamak sayısı, ters çevirme, palindrom, Armstrong; bölen kalıbı: asal ve mükemmel sayı.
- Döngüler 0, 1 ve sınır değerleriyle elle izlenerek test edilmelidir.

ÖZET

Kontrol Soruları

1. while ve do-while arasındaki temel fark nedir?
2. break ile continue ne yapar?
3. Rakamlar toplamında % 10 ve / 10 ne işe yarar?
4. Artan üçgende iç döngünün üst sınırı neden satır değişkenidir?
5. println iç döngüye yazılırsa çıktı nasıl değişir?

ÖZET

Alıştırmalar

1. Klavyeden 0 girilene kadar sayı okuyup toplamlarını ve ortalamalarını yazan programı yazın (while).
2. 1'den 100'e kadar 3'e ya da 5'e bölünen sayıların toplamını bulun (cevap: 2418).
3. n satırlık **piramit** çizin: satır r'de n − r boşluk ve 2r − 1 yıldız (n = 4 için elle izleyin).
4. Bir sayının **faktöriyelini** for ile hesaplayın; 13! neden int'e sığmaz, long ile deneyin.
5. Kullanıcıdan 1–5 arası bir not alana kadar tekrar soran girdi doğrulamasını do-while ile yazın.
6. 1..1000 arasındaki tüm Armstrong (3 basamaklı) ve mükemmel sayıları listeleyin.

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: **while and do-while**, **for**, **Branching Statements**.
- 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

Döngüler

Sonraki sunum: **Metotlar** · **Çalışma soruları** · Simülatör: **Döngü İzleyici** · **Terim sözlüğü**