



# **Bölüm 3: Örnek Çıktılar**

## **JAVA ile Nesne Yönelimli Programlama**

# Örnek



```
void cizdir()  
{  
    System.out.print("*");  
}
```

\*



# Açıklama

- Bu metod, yalnızca bir adet yıldız (\*) karakterini ekrana yazdırır.
- void: Bu metodun geri dönüş değeri olmadığını belirtir.
- cizdir(): Metodun adı. Parametre almaz ve void ile tanımlanır.
- `System.out.print("*")`: Konsola bir adet \* karakteri yazdırır ve imleci aynı satırda tutar.

# Örnek



```
void cizdir(int sayi)
{
    for(int i = 0; i < sayi; i++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

```
* * * * *
* * * * *
```



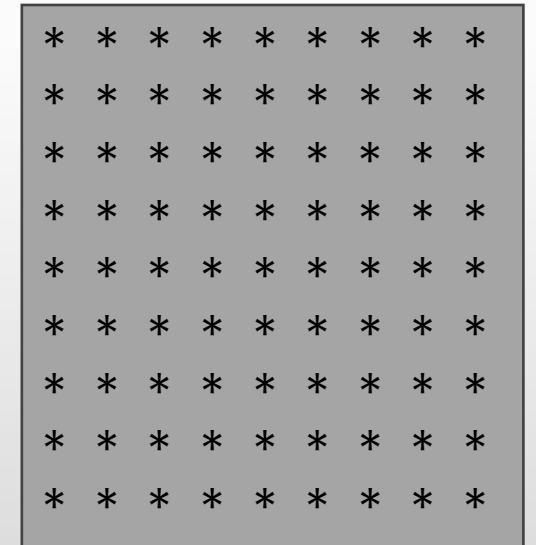
# Açıklama

- Bu metod, girilen sayı kadar yıldız (\*) karakterini tek bir satıra yazdırır.
- `cizdir(int sayi)`: Parametre olarak bir tam sayı (int) alan metod adı.
- `int sayi`: Kullanıcının girdiği, kaç tane \* karakteri yazdırılacağını belirleyen parametre.
- `for(int i = 0; i < sayi; i++)`:
  - Döngü, sayı kadar çalışarak \* karakterini sayı defa yazdırır.
- `System.out.println()`: Döngü sonrasında bir alt satıra geçmek için kullanılır.



# Örnek

```
void cizdir(int sayi)
{
    for(int i = 0; i < sayi; i++)
    {
        for(int j = 0; j < sayi; j++)
        {
            System.out.print("* ");
        }
        System.out.println();
    }
    System.out.println();
}
```





# Açıklama

- Bu metod, belirli bir sayıda satır ve sütun olarak yıldız (\*) karakterlerini bir ızgara (kare) şeklinde yazdırır.
- Dış Döngü (for(int i = 0; i < sayi; i++)):
  - Satır sayısını belirler; her yinelemede yeni bir satır başlatır.
- İç Döngü (for(int j = 0; j < sayi; j++)):
  - Her satırda sayi kadar \* karakterini yan yana yazdırır.
- System.out.print("\*"): Yıldız karakterini (\*) yan yana yazdırır.
- System.out.println(): Her satırın sonunda alt satıra geçilmesini sağlar.



# Örnek

```
void cizdir(int sayi)
{
    for(int i = 0; i < sayi; i++)
    {
        for(int j = 0; j <= i; j++)
        {
            System.out.print("*");
        }
        System.out.println();
    }
    System.out.println();
}
```

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
```





# Açıklama

- Bu metod, bir yıldız (\*) üçgeni oluşturur. Girilen sayi parametresi, üçgenin satır sayısını belirler.
- int sayi: Üçgenin yüksekliğini belirleyen parametre.
- Dış Döngü (for(int i = 0; i < sayi; i++)):
  - Satır sayısını ayarlar. Her yinelemede yeni bir satır oluşturur.
- İç Döngü (for(int j = 0; j <= i; j++)):
  - Her satırda, i değeri kadar \* karakterini yazdırır. Bu, her satırda yıldız sayısını bir artırır.



# Örnek

```
void cizdir(int sayi)
{
    for(int i = 0; i < sayi; i++)
    {
        for(int j = 0; j < sayi - i - 1; j++)
        {
            System.out.print(" ");
        }
        for(int j = 0; j < 2 * i + 1; j++)
        {
            System.out.print("*");
        }
        System.out.println();
    }
    System.out.println();
}
```

```
      *
     ***
    *****
   *********
  ***********
 *****
*****
*****
*****
*****
```



# Açıklama

- Bu metod, yıldız (\*) karakterlerinden bir piramit (üçgen) deseni oluşturur. sayi parametresi, piramidin yüksekliğini (satır sayısını) belirler.
- int sayi: Piramidin yüksekliğini belirleyen parametre.
- Dış Döngü (for(int i = 0; i < sayi; i++)):
  - Satır sayısını belirler. Her yinelemede yeni bir satır başlatır.
- Boşluk Döngüsü (for(int j = 0; j < sayi - i - 1; j++)):
  - Her satırın başında sayi - i - 1 kadar boşluk ekleyerek piramit görünümünü oluşturur.
- Yıldız Döngüsü (for(int j = 0; j < 2 \* i + 1; j++)):
  - Her satırda, tek sayıda yıldız karakterini ortalanmış şekilde yazdırır.



```

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

```



# Açıklama

- Bu metod, ters bir piramit deseni oluşturur. sayi parametresi, piramidin yüksekliğini belirler.
- Boşluk Döngüsü (`for(int j = 0; j < i; j++)`):
  - Her satırın başında i kadar boşluk ekler, böylece yıldızlar ortalanır ve ters piramit görünümü sağlanır.
- Yıldız Döngüsü (`for(int j = 0; j < 2 * (sayi - i) - 1; j++)`):
  - Yıldız karakterlerini yazdırır. Her satırda yıldız sayısı  $2 * (\text{sayi} - i) - 1$  formülüyle azalarak devam eder.



# Örnek

```
void cizdir(int sayi)
{
    for(int i = 0; i < sayi; i++)
    {
        for(int j = 0; j < sayi; j++)
        {
            if(j == 0 || j == sayi-1 || i == 0 || i == sayi-1)
            {
                System.out.print("* ");
            }
            else { System.out.print(" "); }
        }
        System.out.println();
    }
    System.out.println();
}
```

```
* * * * *
*           *
*           *
*           *
*           *
*           *
*           *
*           *
* * * * *
```



# Açıklama

- Bu metod, yıldız (\*) karakterleriyle kenarları çerçeveli bir kare deseni çizer. sayi parametresi, kare çerçevenin yüksekliğini ve genişliğini belirler.
- Koşul (if(j == 0 || j == sayi-1 || i == 0 || i == sayi-1)):
- j == 0 veya j == sayi-1 ise sütunun başında veya sonunda olduğumuz anlamına gelir (sol ve sağ kenarlar).
- i == 0 veya i == sayi-1 ise satırın başında veya sonunda olduğumuz anlamına gelir (üst ve alt kenarlar).
- Bu durumda, \* karakteri yazdırılır.
- Aksi Durum (else): İç kısımlarda ise boşluk karakteri yazdırılır.



# Örnek

```
void cizdir(int sayi)
{
    for(int i = 0; i < sayi; i++)
    {
        for(int j = 0; j < sayi; j++)
        {
            System.out.print(i + j);
        }
        System.out.println();
    }
    System.out.println();
}
```

```
012345678
123456789
2345678910
34567891011
456789101112
5678910111213
67891011121314
789101112131415
8910111213141516
```





# Açıklama

- Bu metod, bir sayı değeri alarak, her bir hücresinde satır ve sütun indislerinin toplamı olan bir kare desen oluşturur.
- Dış Döngü (`for(int i = 0; i < sayi; i++)`):
  - Satır sayısını ayarlar. Her yinelemede yeni bir satır başlatır.
- İç Döngü (`for(int j = 0; j < sayi; j++)`):
  - Her satırdaki sütunları ayarlar.
- `System.out.print(i + j)`: i ve j değerlerini toplar ve sonucu yazdırır.
  - i satır indeksini, j sütun indeksini temsil eder.
  - Toplam değer, o hücreye karşılık gelen sayıyı verir.



# Örnek

```
void cizdir(int sayi)
{
    for(int i = 0; i < sayi; i++)
    {
        for(int j = 0; j < sayi; j++)
        {
            System.out.print(i + " " + j + " ");
        }
        System.out.println();
    }
    System.out.println();
}
```

00	01	02	03	04	05	06	07	08
10	11	12	13	14	15	16	17	18
20	21	22	23	24	25	26	27	28
30	31	32	33	34	35	36	37	38
40	41	42	43	44	45	46	47	48
50	51	52	53	54	55	56	57	58
60	61	62	63	64	65	66	67	68
70	71	72	73	74	75	76	77	78
80	81	82	83	84	85	86	87	88



# Açıklama

- Bu metod, bir sayi parametresi alarak, her hücresinde satır (i) ve sütun (j) indislerini yan yana gösteren bir kare desen oluşturur.
- Dış Döngü (for(int i = 0; i < sayi; i++)):
  - Satır sayısını belirler. Her yinelemede yeni bir satır başlatır.
- İç Döngü (for(int j = 0; j < sayi; j++)):
  - Her satırdaki sütunları yönetir.
- System.out.print(i + "" + j + " "):
  - i ve j değerlerini yan yana yazdırarak her hücrede satır ve sütun indislerini gösterir.
  - i + "" + j ifadesi, i ve j değerlerini birleştirir ve aralarına bir boşluk koyar.



# Örnek

```
void cizdir(int sayi)
{
    for(int i = 1; i <= sayi; i++)
    {
        for(int j = 1; j <= sayi; j++)
        {
            if(i * j < 10) {
                System.out.print("0");
            }
            System.out.print(i * j + " ");
        }
        System.out.println();
    }
    System.out.println();
}
```

01	02	03	04	05	06	07	08	09
02	04	06	08	10	12	14	16	18
03	06	09	12	15	18	21	24	27
04	08	12	16	20	24	28	32	36
05	10	15	20	25	30	35	40	45
06	12	18	24	30	36	42	48	54
07	14	21	28	35	42	49	56	63
08	16	24	32	40	48	56	64	72
09	18	27	36	45	54	63	72	81



# Açıklama

- Bu metod, bir sayı parametresi alarak, çarpım tablosu formatında bir kare desen oluşturur. Her hücre, satır ve sütun indislerinin çarpımını içerir. Tek haneli sonuçlar için başına "0" ekleyerek hizalama sağlanır.
- `if(i * j < 10) { System.out.print("0"); }:`
  - Çarpım sonucu tek haneli ise, başına "0" ekleyerek hizalama sağlar.
- `System.out.print(i * j + " ");:`
  - `i` ve `j` çarpımını hesaplar ve sonuçları her hücrede yan yana yazdırır.



SON