

Çok Boyutlu Diziler

 gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/cok-boyutlu-diziler

Kategori : Mobil | **Lisans :** Creative Commons (by-nc-sa) | En son 28.04.2016 tarihinde güncellendi

Yazar : gy

Etiketler : [java çok boyutlu](#) [java diziler](#)

Çok boyutlu diziler içinde en yaygın kullanım alanı iki boyutlu diziler için söz konusudur.

Elimizde aşağıdaki gibi bir veri mevcut olsun:

1998 Yılı Otomobil Satışları

(Not: Tablodaki rakamlar gerçek değildir, örnek amaçlı verilmiştir.)

MARKA	OCAK	ŞUBAT	MART
Fiat	700	600	650
Renault	900	800	700
VW	300	400	350
Opel	500	450	470
Ford	600	500	480

Bu tabloda, 1998 yılında çeşitli otomobil markaları için gerçekleştirilen ilk üç aylık satışları görüyoruz. Bu tür bir bilgiyi bilgisayara yükleyerek aşağıdaki sorulara cevap olabilecek işlemleri tablo üzerinde gerçekleştirmek isteyebiliriz:

1. Her marka için 3 aylık satış toplamı nedir? (Tablodaki satır toplamı)
2. Her ay için tüm markaların satış toplamı nelerdir? (Tabloda sütun toplamının hesaplanması)
3. Her marka için en çok satışın gerçekleştirildiği ay hangisidir? (Satırlardaki en büyük elemanların tespiti)
4. Her ay için en çok satışın gerçekleştirildiği marka hangisidir? (Sütunlardaki en büyük elemanların tespiti)
5. Tüm marka ve tüm aylar için otomobil satışları toplamı nedir? (Tablonun genel toplamının bulunması)

Tablodaki bilgileri Java dilinde, bilgisayar belleğinde **iki boyutlu dizi (two dimensional array)** yapısı adı verilen yapı içinde saklayabiliriz. Matematikteki *matris* yapılarının Java'daki doğal karşılığı da bu tip *iki boyutlu dizi* yapısıdır.

Yukarıdaki tablo içinde, sayısal olmayan bilgileri ve başlıkları ihmal ederek sadece satışlardan oluşan sayısal bilgiyi dikkate alırsak ve bu bilgiyi *satıs* adlı bir tabloda saklarsak, aşağıdaki yapıyı elde ederiz:

satıs

700	600	650
900	800	700
300	400	350
500	450	470
600	500	480

Java'da, 5 satır ve 3 sütundan oluşan bu tabloyu aşağıdaki şekilde tanımlayabiliriz:

```
int  satis=new int [5,3];
```

Burada satis adlı iki boyutlu dizinin bir elemanına referans vermek için program içinde, şeklinde bir ifade kullanırız. Burada satis dizinin adı, i satır indisi ve j de sütun indisi. Satis dizisinin satır indisleri 0'dan başlar ve 4'e kadar devam eder. Sütun indisleri ise, 0 ile 2 arasında değerler alır. Aşağıda, satis dizisinin bellekte Java tarafından saklanan biçimini görüyoruz:

Buna göre, satis[0,0] elemanı 700, satis[2,1] elemanı 400 ve satis[4,2] elemanı ise 480'dir. Satis adlı iki boyutlu dizinin eleman sayısı ise $5 \times 3 = 15$ 'tir.

İki boyutlu dizinin elemanlarını okutmak

İki boyutlu dizinin elemanlarını Java programının çalıştırılması sırasında klavyeden girmek için aşağıdaki gibi bir program kullanabiliriz. Aşağıdaki örnek programda 3 satır ve 2 sütunu olan satis adlı tablonun elemanları klavyeden giriliyor ve daha sonra tablo ekrana aktarılıyor. Şimdi bu örnek programı inceleyelim:

```

import java.util.Scanner;

public class CokBoyutluDizi1 {
    public static void main(String[] args){
        Scanner giris = new Scanner(System.in);
        int[][] satisTablosu = new int[3][2];
        int i, j;

        /* kullanıcıdan satış tablosu
         * bilgilerini alalım */
        System.out.print("Satis [0,0]=");
        satisTablosu[0][0] = giris.nextInt();

        System.out.print("Satis [0,1]=");
        satisTablosu[0][1] = giris.nextInt();

        System.out.print("Satis [1,0]=");
        satisTablosu[1][0] = giris.nextInt();

        System.out.print("Satis [1,1]=");
        satisTablosu[1][1] = giris.nextInt();

        System.out.print("Satis [2,0]=");
        satisTablosu[2][0] = giris.nextInt();

        System.out.print("Satis [2,1]=");
        satisTablosu[2][1] = giris.nextInt();

        /* bellekteki satis tablosu
         * elemanlarının ekrana yazdırılması */
        System.out.println("Satis tablonuz");
        for (i = 0; i <= 2; i++) {
            for (j = 0; j <= 1; j++) {
                System.out.print(satisTablosu[i][j] + " ");
            }
            System.out.println();
        }
    }
}

```

Programın çıktısı şu şekilde oluşacak:

```

Satis [0,0]=19
Satis [0,1]=29
Satis [1,0]=39
Satis [1,1]=49
Satis [2,0]=89
Satis [2,1]=99
Satis tablonuz
19 29
39 49
89 99

```

İki boyutlu dizi elemanlarına ilk değer atanması

İki boyutlu dizinin elemanları sabit olacaksa yani program her çalıştırıldığında değerleri değişmiyorsa, bu durumda iki boyutlu dizi elemanlarına programın içinde ilk değer ataması yapabiliriz

İlk değer ataması, iki boyutlu dizi değişebilecek olsa bile, dizi üzerinde gerçekleştirilecek tekrarlı bazı işlemler için de bir başlangıç değeri oluşturmak için gerekli olabilir.

satisTablosu adlı dizinin elemanlarını kod bloku içinde ilk değer ataması yoluyla aşağıdaki gibi oluşturabiliriz:

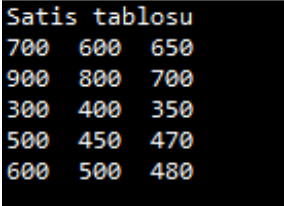
```
int[][] satisTablosu = new int[][]{
    {700,600,650},
    {900,800,700},
    {300,400,350},
    {500,450,470},
    {600,500,480}
};

public class CokBoyutluDizi2 {
    public static void main(String[] args){
        int [][]satisTablosu = new int[][]{
            {700,600,650},
            {900,800,700},
            {300,400,350},
            {500,450,470},
            {600,500,480}};

        int i,j;

        /*bellekteki satis tablosu
         * elemanlarının ekrana yazdırılması*/
        System.out.println("Satis tablosu");
        for(i=0; i<=4; i++){
            for(j=0; j<=2; j++) {
                System.out.print(satisTablosu[i][j]+" ");
            }
            System.out.println();
        }
    }
}
```

Programın çıktısına hep birlikte bakalım:



```
Satis tablosu
700 600 650
900 800 700
300 400 350
500 450 470
600 500 480
```

length fonksiyonunun kullanılması

length fonksiyonu ile iki boyutlu dizinin her bir boyutundaki eleman sayısını bulmamız mümkün.

SATIS			
	0	1	2
0	700	600	650
1	900	800	700
2	300	400	350
3	500	450	470
4	600	500	480

satisTablosu adlı dizi yukardaki gibi tanımlanmışsa *satisTablosu.length* ifadesi 0. boyuta ait eleman sayısını yani satır sayısını (5), *satisTablosu*

.length ifadesiyse 1. boyuta ait eleman sayısını yani sütun sayısını (3) verecektir. Aşağıdaki Java kodu, bu fonksiyonların nasıl kullanılabileceğini gösteriyor:

```
public class CokBoyutluDizi3 {
    public static void main(String[] args){
        int [][]satisTablosu = new int[][]{
            {700,600,650},
            {900,800,700},
            {300,400,350},
            {500,450,470},
            {600,500,480}
        };
        int i,j;

        /*bellekteki satis tablosu
        * elemanlarının ekrana yazdırılması*/
        System.out.println("Satis tablosu");
        /*
        * döngülerin sınırlarını length()
        * fonksiyonlarıyla belirliyoruz
        * */
        for(i=0; i<satisTablosu.length; i++) {
            for(j=0; j<satisTablosu[0].length; j++) {
                System.out.print(satisTablosu[i][j]+"  ");
            }
            System.out.println();
        }
    }
}
```

Program çıktısı:

ÖRNEK: Satış Tablosunun Satır ve Sütun Toplamlarının ve Tablodaki Elemanların Genel Toplamının Bulunması

```
Satis tablosu
700 600 650
900 800 700
300 400 350
500 450 470
600 500 480
```

```

public class CokBoyutluDizi4 {
    public static void main(String[] args){
        int [][]satisTablosu = new int[][]{
            {700,600,650},
            {900,800,700},
            {300,400,350},
            {500,450,470},
            {600,500,480}
        };
        int i, j, toplamSatis, toplamOtomobilSayisi=0;

        /*bellekteki satisTablosu dizisinin
        * elemanlarının ekrana yazdırılması*/
        System.out.println("Otomobil satislari tablosu");
        for(i=0; i<satisTablosu.length; i++) {
            for(j=0; j<satisTablosu[0].length; j++) {
                System.out.print(satisTablosu[i][j]+" ");
            }
            System.out.println();
        }

        /*
        * Her marka için 3 aylık satış toplamalarını
        * yani dizinin satır toplamalarını yazdıralım
        * */
        for(i=0; i<=4; i++) {
            toplamSatis=0;
            for(j=0; j<=2; j++)
                toplamSatis=toplamSatis+satisTablosu[i][j];
            System.out.println(i+1+". markanın toplam satisi: "+toplamSatis);
        }

        /*
        * Her ay için 5 markanın satış toplamalarını
        * yani tablodaki sütunların toplamalarını
        * gösterelim
        * */
        System.out.println();
        toplamOtomobilSayisi=0;
        for(j=0; j<=2; j++){
            toplamSatis=0;
            for(i=0; i<=4; i++)
            {
                toplamSatis=toplamSatis+satisTablosu[i][j];
                toplamOtomobilSayisi=toplamOtomobilSayisi+satisTablosu[i][j];}
            System.out.println(j+1+". ayda 5 marka için toplam satis miktarı:
"+toplamSatis);
        }
        /*3 AY ICIN TOPLAM OTOMOBIL SATISI-GENEL TOPLAM*/
        System.out.println("3 aylık toplam otomobil satisi"+toplamOtomobilSayisi);
    }
}

```

Program çıktısı:

```

Otomobil satislari tablosu
700 600 650
900 800 700
300 400 350
500 450 470
600 500 480
1. markanin toplam satisi: 1950
2. markanin toplam satisi: 2400
3. markanin toplam satisi: 1050
4. markanin toplam satisi: 1420
5. markanin toplam satisi: 1580

1. ayda 5 marka icin toplam satis miktarı: 3000
2. ayda 5 marka icin toplam satis miktarı: 2750
3. ayda 5 marka icin toplam satis miktarı: 2650
3 aylık toplam otomobil satisi: 8400

```

ÖRNEK: Aşağıda verilen matrisi, bilgisayarda otomatik olarak oluşturacak bir Java programı yazalım.

2!	4!	6!
8!	10!	12!
14!	16!	18!

```

public class CokBoyutluDizi5 {
    public static void main(String[] args){
        double[][] dizi=new double[3][3];
        int i,j,sayimiz;
        sayimiz=2;
        for(i=0; i<=2; i++) {
            for(j=0; j<=2; j++) {
                dizi[i][j]=faktoriyel(sayimiz);
                System.out.println(sayimiz+"!="+dizi[i][j]);
                sayimiz=sayimiz+2;
            }
        }
    }

    static double faktoriyel(int n){
        double fakt;
        int i;
        fakt=1.0;
        for(i=1; i<=n; i++) {
            fakt=fakt*i;
        }
        return fakt;
    }
}

```

Programın çıktısı şöyle olacaktır:

ÖRNEK: Önceden belirleyeceğimiz 3'e 4'lük bir tabloda yer alan elemanların en küçük ve en büyük değerdeki elemanı bulan bir Java programı yazalım. Ayrıca bu program her satırdaki ve sütundaki en küçük ve en büyük elemanları da bulabilsin.

```
public class CokBoyutluDizi6 {
    public static void main(String[] args){

        double[][] tablo=new double [][]{
            {34,11,-7,53},
            {254,8,-65,4},
            {110,64,33,26}
        };
        int i,j;
        double kucukEleman, buyukEleman, enKucuk, enBuyuk;

        /*tablonun yazdırılması*/
        System.out.println("Dizinin Elemanlari");
        for(i=0; i<=2; i++) {
            for (j=0; j<=3; j++) {
                System.out.print(tablo[i][j]+"\\t");
            }
            System.out.println();
        }

        /* Her satırdaki en büyük ve en küçük
        * elemanların bulunması*/
        for(i=0; i<=2; i++) {
            kucukEleman=3.4e38;
            buyukEleman=-3.4e38;
            for(j=0; j<=3; j++) {
                if(tablo[i][j]<kucukEleman) {
                    kucukEleman=tablo[i][j];
                }
                if(tablo[i][j]>buyukEleman) {
                    buyukEleman=tablo[i][j];
                }
            }
            System.out.println((i+1)+"\\n. satirdaki en kucuk eleman: "+kucukEleman);
            System.out.println((i+1)+"\\n. satirdaki en buyuk eleman: "+buyukEleman);
        }

        /* Her sütundaki en büyük ve en küçük
        * elemanın bulunması*/

        /* Ek olarak tablodaki en büyük ve en küçük
        * elemanın bulunması */
        enKucuk=3.4e38;
        enBuyuk=-3.4e38;
        for(j=0; j<=3; j++) {
            kucukEleman=3.4e38;
            buyukEleman=-3.4e38;
            for (i=0; i<=2; i++) {
                if(tablo[i][j]<kucukEleman) {
                    kucukEleman=tablo[i][j];
                }
            }
        }
    }
}
```

```
2!=2.0
4!=24.0
6!=720.0
8!=40320.0
10!=3628800.0
12!=4.790016E8
14!=8.71782912E10
16!=2.0922789888E13
18!=6.402373705728E15
```

```

    }
    if(tablo[i][j]>buyukEleman) {
        buyukEleman=tablo[i][j];
    }
    if(tablo[i][j]<enKucuk) {
        enKucuk=tablo[i][j];
    }
    if(tablo[i][j]>enBuyuk) {
        enBuyuk=tablo[i][j];
    }
}
System.out.println((j+1)+". sutundaki en kucuk eleman: "+kucukEleman);
System.out.println((j+1)+". sutundaki en buyuk eleman: "+buyukEleman);
}
System.out.println("Tablodaki en kucuk eleman: "+enKucuk);
System.out.println("Tablodaki en buyuk eleman: "+enBuyuk);
}
}
}

```

Programın çıktısı:

```

Dizinin Elemanlari
34.0    11.0    -7.0    53.0
254.0   8.0    -65.0   4.0
110.0   64.0   33.0   26.0
1. satirdaki en kucuk eleman: -7.0
1. satirdaki en buyuk eleman: 53.0
2. satirdaki en kucuk eleman: -65.0
2. satirdaki en buyuk eleman: 254.0
3. satirdaki en kucuk eleman: 26.0
3. satirdaki en buyuk eleman: 110.0
1. sutundaki en kucuk eleman: 34.0
1. sutundaki en buyuk eleman: 254.0
2. sutundaki en kucuk eleman: 8.0
2. sutundaki en buyuk eleman: 64.0
3. sutundaki en kucuk eleman: -65.0
3. sutundaki en buyuk eleman: 33.0
4. sutundaki en kucuk eleman: 4.0
4. sutundaki en buyuk eleman: 53.0
Tablodaki en kucuk eleman: -65.0
Tablodaki en buyuk eleman: 254.0

```

ÖRNEK: Matris Çarpımı

Aşağıdaki Java programı,

$$C_{mn} = A_{mk} B_{kn}$$

şeklindeki matris çarpımı işlemini gerçekleştiriyor:

```

public class CokBoyutluDizi7 {
    public static void main(String[] args){

        int[][] aMatris=new int [][]{{2,1},{-1,4},{5,3}};
        int[][] bMatris=new int [][]{{3,2,1,-1},{4,-2,1,2}};
        int[][] cMatris=new int [3][4];
        int i,j,k,t;

        /*a ve b matrislerinin yazdırılması*/
        System.out.println("A matrisi");
        for(i=0; i<=2; i++) {
            for(j=0; j<=1; j++) {
                System.out.print(aMatris[i][j]+"\\t");
            }
            System.out.println();
        }

        System.out.println("B matrisi");
        for(i=0; i<=1; i++){
            for(j=0;j<=3;j++) {
                System.out.print(bMatris[i][j]+"\\t");
            }
            System.out.println();
        }

        /* a ve matrisinin arpımı olan
        * c matrisinin hesaplanması */
        for(k=0; k<=3; k++) {
            for(i=0; i<=2; i++) {
                cMatris[i][k]=0;

                for(j=0; j<=1; j++) {
                    cMatris[i][k]=cMatris[i][k]+
                        aMatris[i][j]*bMatris[j][k];
                }
            }
        }
        System.out.println("C matrisi");
        for(i=0; i<=2; i++)
        {
            for(j=0;j<=3;j++)
                System.out.print(cMatris[i][j]+"\\t");
            System.out.println();
        }
    }
}

```

Programın ıktısı:

ÖRNEK: Aşağıdaki matrisi otomatik olarak oluşturan ve yazdıran bir Java programı geliştirelim.

$(2! - 1) / 3$	$(4! + 2) / 5$	$(6! - 3) / 7$
$(8! + 4) / 9$	$(10! - 5) / 11$	$(12! + 6) / 13$

```
A matrisi
2      1
-1     4
5      3
B matrisi
3      2      1      -1
4      -2     1      2
C matrisi
10     2      3      0
13     -10    3      9
27     4      8      1
```

```

public class CokBoyutluDizi8 {
    public static void main(String[] arg){
        double matris[][]=new double[2][3];
        int i,j,p,u;
        int k;
        char isaret;
        p=-1;
        k=2;
        u=1;
        for(i=0; i<=1; i++) {
            for(j=0; j<=2; j++) {
                matris[i][j]=(faktoriyel(k)+p*u)/(k+1.0);
                if(p<0) {
                    isaret='-';
                }
                else {
                    isaret='+';
                }
                System.out.println("[ "
                    +i+
                    "]"+"
                    +j+
                    "] = ( "
                    +k+
                    "! "
                    +isaret+u+
                    ") / "
                    +(k+1)+
                    " = "
                    +matris[i][j]+
                    "\t");

                k=k+2;
                p=-p;
                u++;
            }
        }
    }

    public static double faktoriyel(double sayi){
        double fakt=1.0;
        int i;
        for(i=1; i<=sayi; i++) {
            fakt=fakt*i;
        }
        return fakt;
    }
}

```

Programın çıktısı şu şekilde olacaktır:

ÖRNEK: Aşağıdaki matrisin **en büyük** elemanının bulunduğu sütunda yer alan **en küçük** elemanı bulan bir Java programı geliştirelim.

56	23	678	231
234	21	78	23
654	33	22	67
189	35	56	89

```
A matrisi
2      1
-1     4
5      3
B matrisi
3      2      1      -1
4      -2     1      2
C matrisi
10     2      3      0
13     -10    3      9
27     4      8      1
```

Yazılacak olan Java programı doğru çalıştığı takdirde, tablodaki en büyük elemanı 678 olan sütuna gelip oradaki en küçük sayı olan 22'yi bulmalıdır.

```
public class CokBoyutluDizi9 {
    public static void main(String[] arg){
        double[][] matris={
            {56,23,678,231},
            {234,21,78,23},
            {654,33,22,67},
            {189,35,56,89}
        };
        int i,j,k;
        k=0;
        double enBuyuk,enKucuk;
        enBuyuk=matris[0][0];
        for(i=0; i<=3; i++){
            for(j=0; j<=3; j++){
                if(matris[i][j]>enBuyuk)
                {
                    enBuyuk=matris[i][j];
                    k=j;
                }
                System.out.print("[ "+i+" ][ "+j+" ]=" + matris[i][j]+"\\t");
            }
            System.out.println();
        }
        enKucuk=matris[0][k];
        for(i=0; i<=3; i++) {
            if(matris[i][k]<enKucuk) {
                enKucuk=matris[i][k];
            }
        }
        System.out.println("En buyuk elemanin bulundu\u011fu sutundaki en kucuk
sayi"+enKucuk);
    }
}
```

Programın çıktısını aşağıda bulabilirsiniz:

```
[0][0]=56.0    [0][1]=23.0    [0][2]=678.0    [0][3]=231.0
[1][0]=234.0    [1][1]=21.0    [1][2]=78.0    [1][3]=23.0
[2][0]=654.0    [2][1]=33.0    [2][2]=22.0    [2][3]=67.0
[3][0]=189.0    [3][1]=35.0    [3][2]=56.0    [3][3]=89.0
En buyuk elemanın bulunduğu sutundaki en kucuk sayi22.0
|
```

ÖRNEK: Aşağıdaki matrisin içindeki **en küçük ikinci** elemanı bulan bir Java programı geliştirelim.

56	23	678	231
234	21	78	26
654	33	32	67
189	35	56	89

Program, bu tablo içindeki en küçük ikinci eleman olan 23'ü bulmalıdır.

```

public class CokBoyutluDizi10 {
    public static void main(String[] arg){
        double[][] matris={
            {56,23,678,231},
            {234,21,78,26},
            {654,33,32,67},
            {189,35,56,89}
        };
        int i,j,y,x;
        y=0;
        x=0;
        double enKucuk;
        enKucuk=matris[0][0];
        for(i=0; i<=3; i++) {
            for(j=0; j<=3; j++){
                if(matris[i][j]<enKucuk) {
                    enKucuk=matris[i][j];
                    x=i;
                    y=j;
                }
                System.out.println "["+i+"]"+"["+j+"] = "+matris[i][j]+"\\t");
            }
        }
        matris[x][y]=1000000000000.0;
        enKucuk=matris[0][0];
        System.out.println("");
        for(i=0; i<=3; i++){
            for(j=0; j<=3; j++){
                if(matris[i][j]<enKucuk)
                {
                    enKucuk=matris[i][j];
                    y=j;
                    x=i;
                }
                System.out.print "["+i+"]"+"["+j+"]=" + matris[i][j]+"    ");
            }
            System.out.println("");
        }
        System.out.println("");
        System.out.println("");
        System.out.println("Matristeki ikinci en kucuk eleman: "+enKucuk);
        System.out.println("Bu elemanın satır ve sütunu "+x+"'a "+y);
    }
}

```

Programın çıktısı:

```

[0][0] = 56.0
[0][1] = 23.0
[0][2] = 678.0
[0][3] = 231.0
[1][0] = 234.0
[1][1] = 21.0
[1][2] = 78.0
[1][3] = 26.0
[2][0] = 654.0 |
[2][1] = 33.0
[2][2] = 32.0
[2][3] = 67.0
[3][0] = 189.0
[3][1] = 35.0
[3][2] = 56.0
[3][3] = 89.0

[0][0]=56.0  [0][1]=23.0  [0][2]=678.0  [0][3]=231.0
[1][0]=234.0  [1][1]=1.0E11  [1][2]=78.0  [1][3]=26.0
[2][0]=654.0  [2][1]=33.0  [2][2]=32.0  [2][3]=67.0
[3][0]=189.0  [3][1]=35.0  [3][2]=56.0  [3][3]=89.0

Matristeki ikinci en kucuk eleman: 23.0
Bu elemanin satir ve sutunu 0'a 1

```

ÖRNEK: Aşağıdaki matrisin içindeki tek sayıları tek boyutlu bir diziye yükleyen ve bu diziyi sıralayan bir Java programı geliştirelim.

56	23	678	231
>234	21	78	26
654	33	32	67
189	35	56	89

Tablodaki tek sayılar koyu hale getirilmiştir. Yazılacak program bu sayıları tek boyutlu diziye yükleyerek sıralamalıdır.

```

public class CokBoyutluDizi11 {
    public static void main(String[] arg){
        int[][] matris={
            {56,23,678,231},
            {234,21,78,26},
            {654,33,32,67},
            {189,35,56,89}
        };
        int i,j,k;
        k=0;
        int temp=0;
        int[] tekSayilar=new int[16];

        /*tek sayıları bulalım*/
        for(i=0; i<=3; i++) {
            for(j=0; j<=3; j++) {
                if (matris[i][j]/2*2 != matris[i][j]) {
                    tekSayilar[k]=matris[i][j];
                    k++;
                }
                System.out.print "["+i+"["+j+"]=" + matris[i][j]+"\\t");
            }
            System.out.println();
        }

        System.out.println();
        System.out.println("Tek sayılardan oluşan dizi:");
        for(i=0; i<=k-1; i++) {
            System.out.print(tekSayilar[i]+"\\t");
        }
        System.out.println();
        System.out.println();

        /*tek sayılardan oluşan diziyi küçükten
        * büyüğe olacak şekilde sıralayalım*/
        for(i=0; i<tekSayilar.length; i++) {
            for(j=0; j<tekSayilar.length; j++) {
                if(tekSayilar[i]>tekSayilar[j]) {
                    temp=tekSayilar[i];
                    tekSayilar[i]=tekSayilar[j];
                    tekSayilar[j]=temp;
                }
            }
        }
        System.out.println("Tek sayılardan oluşan dizinin sıralanmış hali:");
        for(i=k-1; i>=0; i--) {
            System.out.print(tekSayilar[i]+"\\t");
        }
    }
}

```

Programın çıktısı:

```
[0][0]=56      [0][1]=23      [0][2]=678     [0][3]=231
[1][0]=234     [1][1]=21      [1][2]=78      [1][3]=26
[2][0]=654     [2][1]=33      [2][2]=32      [2][3]=67
[3][0]=189     [3][1]=35      [3][2]=56      [3][3]=89

Tek sayılardan oluşan dizi:
23      231      21      33      67      189      35      89

Tek sayılardan oluşan dizinin sıralanmış hali:
21      23      33      35      67      89      189      231
```

ÖRNEK: Alfabetik karakterlerden oluşan aşağıdaki matris veriliyor:

m	a	v	b
z	y	c	e
g	f	p	q
r	h	i	k

Bu matristeki **c-r** arasındaki harfleri tek boyutlu bir diziye yerleştirerek bu diziyi sıralayan bir Java programı geliştirelim.

```

public class CokBoyutluDizi12 {
    public static void main(String[] arg){
        char[][] matris={
            {'m','a','v','b'},
            {'z','y','c','e'},
            {'g','f','p','q'},
            {'r','h','i','k'}
        };

        int i,j,k;
        k=0;
        char temp;
        char[] harfler=new char[16];

        /*dizideki harflerden c-r arasınd
        * olanları bulalım */
        for(i=0; i<=3; i++){
            for(j=0; j<=3; j++) {
                if(matris[i][j]>='c' && matris[i][j]<='r') {
                    harfler[k]=matris[i][j];
                    k++;
                }
                System.out.print "["+i+"["+j+"]=" + matris[i][j]+"\\t");
            }
            System.out.println();
        }

        /*c-r arası harfleri ekrana basalım*/
        System.out.println();
        System.out.println("c-r arası harfler: ");
        for(i=0; i<=k-1; i++) {
            System.out.print(harfler[i]+" ");
        }

        /*c-r arasındaki harfleri sıralayalım*/
        for(i=0; i<harfler.length; i++){
            for(j=0; j<harfler.length; j++){
                if(harfler[i]>harfler[j]) {
                    temp=harfler[i];
                    harfler[i]=harfler[j];
                    harfler[j]=temp;
                }
            }
        }

        System.out.println();
        System.out.println("Sıralanmış halleri: ");
        /*sıralanan harfleri ekrana basalım*/
        for(i=k-1; i>=0; i--) {
            System.out.print(harfler[i]+" ");
        }
    }
}

```

Programın çıktısı:

```
[0][0]=m      [0][1]=a      [0][2]=v      [0][3]=b
[1][0]=z      [1][1]=y      [1][2]=c      [1][3]=e
[2][0]=g      [2][1]=f      [2][2]=p      [2][3]=q
[3][0]=r      [3][1]=h      [3][2]=i      [3][3]=k

c-r arasi harfler:
m c e g f p q r h i k
Siralanmis halleri:
c e f g h i k m p q r
```

Android uygulamalarında, özellikle uzaktaki bir kaynaktan ya da veri tabanından çoklu veri çektiğinizde dizi yapılarını kullanıyor olacağız.

[Önceki Sayfa](#) [Sonraki Sayfa](#)