

NESNE YÖNELİMLİ PROGRAMLAMA(Object Oriented Programming/OOP)

Öğr. Gör. Celil ÖZTÜRK

Marmara Üniversitesi

Teknik Bilimler Meslek Yüksekokulu

İçerik

- ✓ Tek Boyutlu Diziler
- ✓ Çok Boyutlu Diziler
- ✓ Bellek Yönetimi(Memory Management)
 - Stack Alanı
 - Heap Alanı
- ✓ String Kelimesi !!

Tek Boyutlu Diziler(One Dimentional Array)

- Dizi(Array) bir veri yapısı çeşididir.
- Diziler aynı tip veri topluluklarını saklamak için kullanılır.
- Bilgisayar belleğinde aynı tipte verilerin, tek bir isim altında tutulduğu yapıdır.
- Dizinler yani Arrays sınıfı java.util paketi içinde Object sınıfının bir alt sınıfıdır.

java.lang.Object sınıfından türetildiği için, Object sınıfının tüm özelliklerine sahiptir ve 110'dan fazla bu sınıfa ait metod vardır.

Tek Boyutlu Diziler(One Dimensional Array)

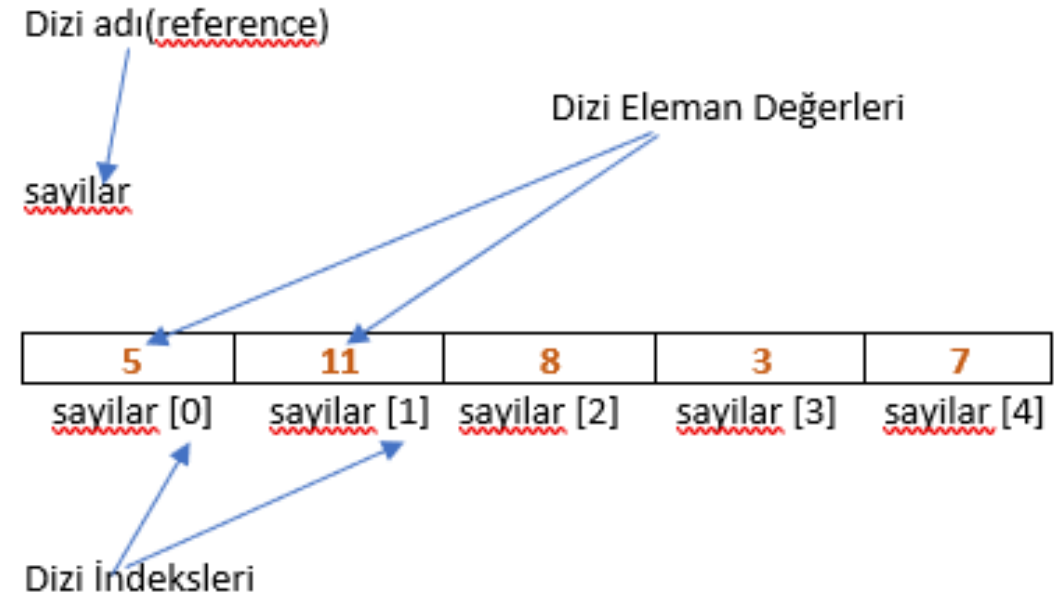
- İndisler negatif olamaz.
- İndisler tamsayı veya tamsayı üreten bir değer olmalıdır.
- Dizi tanımlama: Tip=Veri Tipi
 - Veritipi [] diziAdi = new veritipi [DiziElemanSayısı]
 - int [] sayilar= new int [10];
- Yukarıda int türünde sayilar adında 10 elemanlı bir dizi oluşturulmaktadır.

Tek Boyutlu Diziler

```
int [ ] sayilar = new int [5];
```



- Dizi elemanlarına erişim:
- `sayilar[0]=5;` //Dizide belirtilen indekse değer yazma
- `System.out.println(sayilar[1])`
- // Yukarıdaki Kod çıktısı "11"



Tek Boyutlu Diziler

Dizi tanımlama:

```
String [] dersler= {"Veritabanı","Nesne Tabanlı Programlama","Makine Öğrenmesi"};  
System.out.println(dersler[2]);
```

```
int []sayilar2 ={1,5,9};  
System.out.println(sayilar2[0]);
```

Dizi yazdırma:

```
for(String eleman : dersler)  
{  
    System.out.println(eleman);  
}
```

```
for(int eleman : sayilar)  
{  
    System.out.println(eleman);  
}
```

Çok Boyutlu Diziler

İki Boyutlu Dizi

Veritipi [] [] diziAdi = new Veritipi [satirSayisi] [sutunSayisi];

int [] [] ikiboyutludizi = new int[2][3];

Matris [0] [0]	Matris [0] [1]	Matris [0] [2]	Matris [0] [m]
Matris [1] [0]	Matris [1] [1]	Matris [1] [2]	Matris [1] [m]
....
....
Matris [n] [0]	Matris [n] [1]	Matris [n] [2]	Matris [n] [m]




Çok Boyutlu Diziler

```
int [ ][ ] cokboyutludizi = new int[2][3];  
cokboyutludizi[0][0]=20;  
cokboyutludizi[0][1]=40;  
cokboyutludizi[0][2]=60;
```


```
cokboyutludizi[1][0]=10;  
cokboyutludizi[1][1]=30;  
cokboyutludizi[1][2]=50;
```

```
for(int i=0;i<2;i++)  
{ for(int k=0;k<3;k++)  
{  
    System.out.print "["+cokboyutludizi[i][k]+ " ] ";  
}  
    System.out.println();  
}
```

2x3 İki Boyutlu Dizi Matrisi



20	40	60
10	30	50



```
Output - JavaApplication1 (run) X  
run:  
[20] [40] [60]  
[10] [30] [50]  
BUILD SUCCESSFUL (total time: 1 second)  
>>
```


Dizi Uzunluğu

- Bir dizinin eleman sayısını (boyutunu) bulmak için length metodu kullanılır. Bu metodun döndürdüğü değer int tipindedir.

- Örnek:

```
int A[ ]={8,9,12,4,5,7,1,0,6};
```

şeklinde tanımlanan bir dizinin boyutunu (eleman sayısını) öğrenmek için A.length komutunu kullanırız.

DİZİ ve İNDİSLERE İSİM VERİRKEN DİKKAT EDİLECEK NOKTALAR

- Dizi tanımlama esnasında indis değeri verilmez (boyutu belirtilmez) ise dizinin boyutu tanımlanan eleman sayısı kadardır.

- Örnek:

```
int A [ ]={1, 2, 3, 4, 5};
```

- Şeklindeki tanımlamada dizinin boyutu 5'dir.
- Dizinin ilk indis değeri 0 (sıfır)'dır. Dolayısıyla N elemanlı bir dizinin N. elemanının indis değeri N-1'dir.

Dizi Sıralama(Array.sort)

- Dizilerde sıralama yapmak için `Array.sort()`; komutu kullanılır.
- `import java.util.Arrays` sınıfı projeye eklenmelidir.

```
int [] teksayilar={3,9,7,1,5};

for( int i=0;i<5;i++)
{
    System.out.print(" "+teksayilar[i]); //Dizi ekrana basıldı.
}
System.out.println();

Arrays.sort(teksayilar); // Dizi sıralandı.
for( int i=0;i<teksayilar.length;i++)
{
    System.out.print(" "+teksayilar[i]); //Sıralanan dizi ekrana basıldı
}
```

```
for (int i = 0; i < teksayilar.length-1; i++)
{
    for (int j = i+1; j < teksayilar.length; j++)
    {
        if (teksayilar[i]>teksayilar[j]) {
            int tmp=teksayilar[i];
            teksayilar[i]=teksayilar[j];
            teksayilar[j]=tmp;
        }
    }
}
```

NOT:“diziadi.length” komutu dizinin uzunluğunu verir.

Dizilerde Arama(Arrays.binarySearch)

- Dizilerde arama için, Arrays.binarySearch(); komutu kullanılır.
- BinarySearch(diziadi,aranacakarguman);

```
int [] teksayilar={3,9,7,1,5,11,8};
```

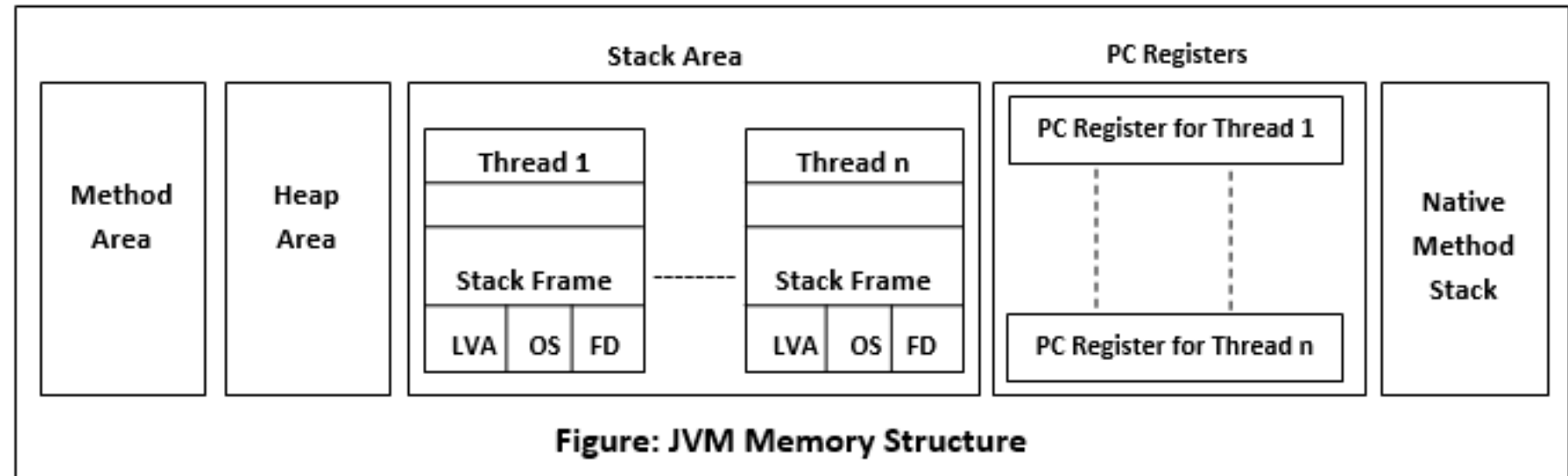
```
int retVal=Arrays.binarySearch(teksayilar,11);
```

```
System.out.println("11 elemanının indeksi:"+retVal);
```

Java Bellek Yönetimi(Memory Management)

Javada bellek yönetimi JVM ve Garbage Collector(Çöp Toplayıcı) ile yapılır. JVM ve GC ile memory üzerinde alan tahsis(memory allocation) ve serbest bırakma(deallocation) işlemleri yapılır.

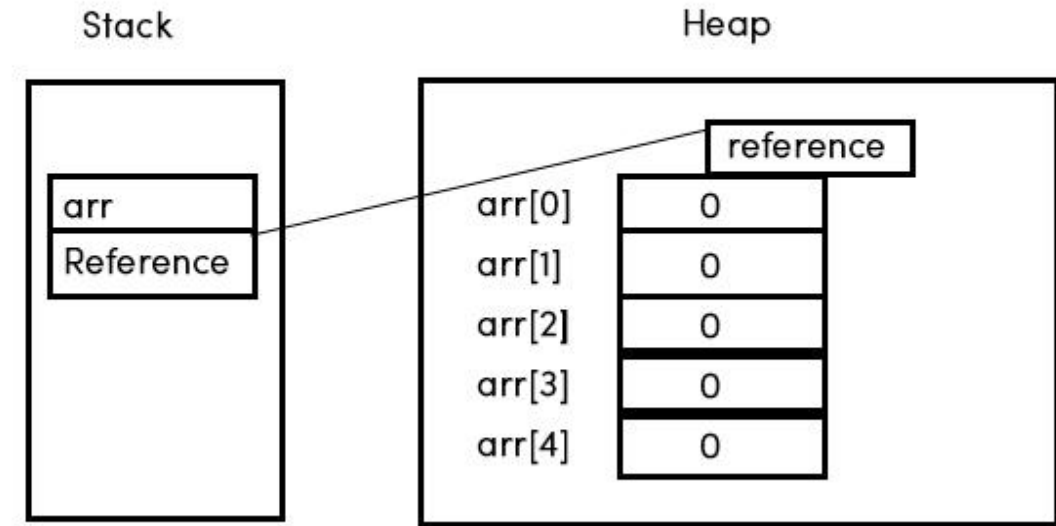
Garbage Collector otomatik bellek yönetimi yapan bir yapıdır. Kullanılmayan alanlar GC tarafından boşaltılır.



Java Bellek Yönetimi(Memory Management)

- Yandaki görselde, int türünde 5 elemanlı "arr" adında bir dizi tanımlanmıştır.
- Diziler bellekte iki farklı alanda saklanır.
- Dizinin adı, yani referansı Stack yapısında, Dizinin elemanları ise Heap alanında tutulur.
- Diğer bir ifade ile, değişebilen değerler Heap, değişmeyen değerler ise Stack alanında depolanır.
- Stack alanında referans değişkenler ve lokal değişkenler yer alır.

```
int[] arr = new int[5];
```



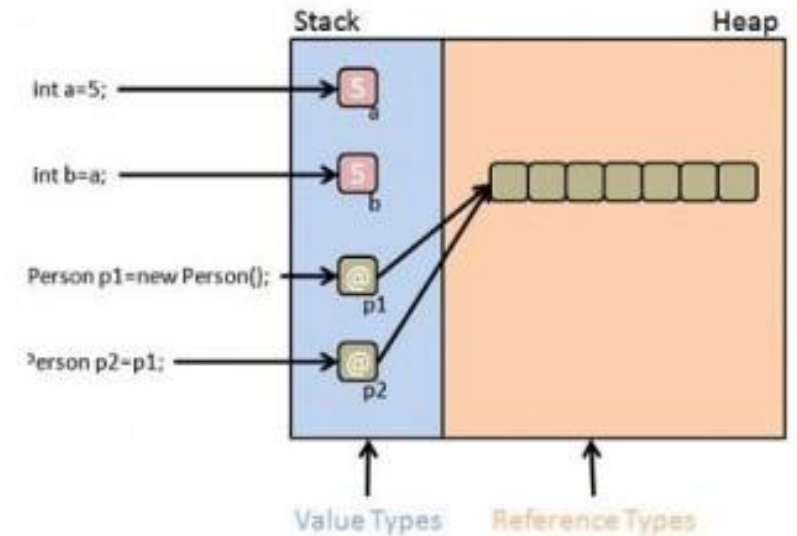
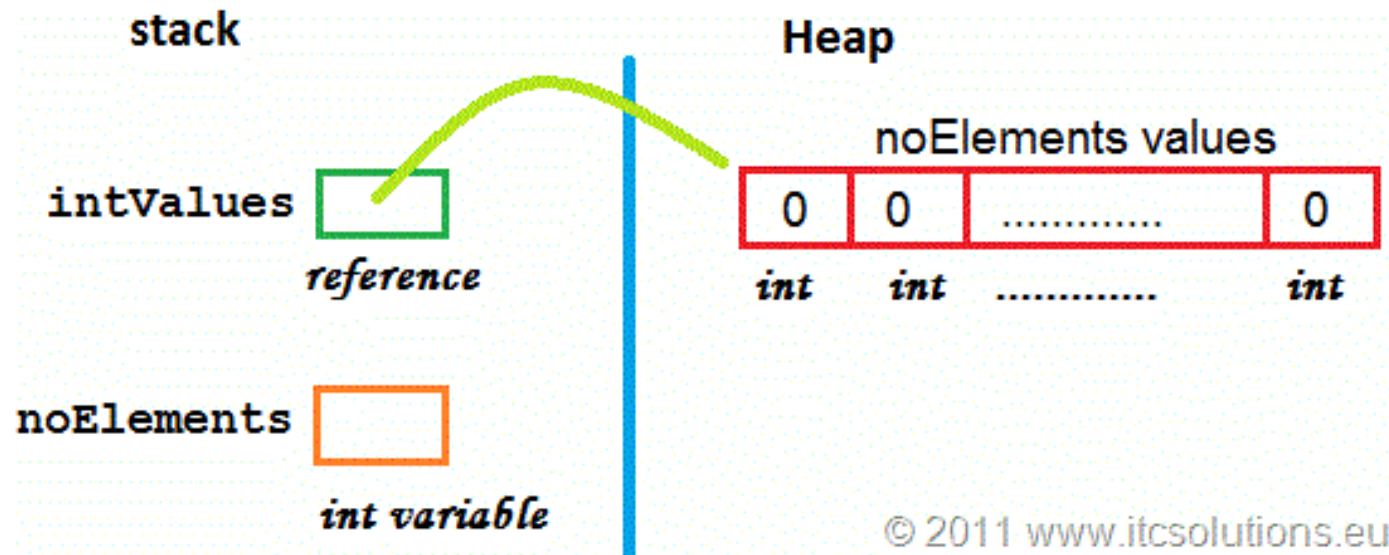
Java Bellek Yönetimi(Memory Management)

Referans Tip(Reference Type)

Sınıf, Diziler, Arayüz(Interface), !String

Değer Tip(Value Type)

int, short, long, boolean, double...



Çok Boyutlu Dizi

Kullanım şekli:

Tip dizi_adı [] [] [] = new Tip [satır_boyutu] [sütun boyutu] [Matris sayısı];

3 boyutlu A dizisini

`int A[][][] = new int [4][5][3];`

şeklinde tanımlarız.

Bu ifade “4 satır, 5 sütunu olan 3 adet iki boyutlu matris için yer aç” şeklinde okunur. A dizisi $4 \cdot 5 \cdot 3 = 60$ elemana sahiptir.

Dizi Elemanlarının Başka Diziye Kopyalanması

Java dilinde dizi elemanlarını başka bir diziye kopyalamak için `System.arraycopy()` yöntemi kullanılır. Bu yöntemin kullanım şekli;

`System.arraycopy(a, başlangıç, b, başlangıç, miktar);`

Bu komut satırı, miktar ile belirtilen sayıdaki elemanı, a dizisinin başlangıç elemanından itibaren, b dizisine başlangıç elemanından başlayarak kopyalar.

Dizi Elemanlarının Başka Diziye Kopyalanması

Örnek a dizisinin elemanlarını, b dizisine kopyalayan ve ekrana tersten yazan programı yazınız. (a dizisi {"ali","betül","gani","zehra"})

```
public class Dizix {  
    public static void main(String[] args) {  
        String [ ]a={"ali","betül","gani","zehra"};  
        String [ ]b=new String[a.length];  
        System.arraycopy(a,0,b,0,a.length);  
        for(int i=a.length-1;i>=0;i--){  
            System.out.print(b[i]+"\\t");  
        }  
    }  
}
```

STRING İŞLEMLERİ

Stringlerle ilgili açıklamalar:

- String'leri ilk yapılandırırken new deyimini kullanmanız gerekmez. Java otomatik olarak String nesnesini yapılandırır, yani “ ” işaretleri arasında kalan ifadeyi ilk değer olarak atar. Aşağıdaki iki tanımlama da aynı işlevi yapar:
 String Ad="ali";
 String Ad=new String("ali");
- String ifadeleri birleştirmek için + operatörü kullanılır. Örneğin;
 System.out.println("Ada"+"Pazarı"); //Ekrana AdaPazarı yazar.
- String ifadenin karakter uzunluğunu öğrenmek için **length()** komutu kullanılır. Örneğin;
- String ad1=""; ad1.length()→0
- String ad2="Selam" ad2.length() →5

STRING İŞLEMLERİ

String İşlemlerle İlgili Metotlar

String ifadeden karakter elde etmek için;

1. `charAt(indis)`: String ifadede `indis` pozisyonundaki harfi geri döndürür. İlk karakterin pozisyonu 0'dır.
2. `substring(Başlangıç_indisi)`: String ifadede `Başlangıç_indisi` pozisyonundan başlayan ve sona kadar devam eden alt string ifadeyi geri döndürür.
3. `substring(Başlangıç_indisi, Bitiş_indisi)`: String ifadede `Başlangıç_indisi` pozisyonundan başlayan ve `Bitiş_indisi` pozisyonuna kadar (dahil değil) devam eden alt string ifadeyi geri döndürür.

Tek Boyutlu Dizi Uygulama

- Örnek:
Klavyeden girilen 6 adet sayıyı, Sayi isimli diziye aktarıp, ekranda gösteren programı yazınız.

```
import java.util.Scanner;
public class DiziyeEkleme {
    public static void main(String[] args) {
        int[] Sayi=new int[6];
        int veri;
        Scanner scan=new Scanner (System.in);
        //dizi elemanlarına veri girişi yapılıyor
        for(int i=0; i<Sayi.length; i++)
        {System.out.println("Tamsayı giriniz: ");
        veri=scan.nextInt();
        Sayi[i]=veri;
        }
        //dizi elemanları ekranda gösteriliyor
        for(int i=0; i<Sayi.length; i++)
        {System.out.println("Sayi[" + i + "]=" + Sayi[i]);
        }
    }
}
```

Tek Boyutlu Dizi Uygulama

Örnek:

Girilen rakam değerine göre yılın hangi ayı olduğunu hesaplayıp ekranda gösteren programı yazınız.

(Ay değeri 1 ile 12 arasında girilmezse kullanıcıya uyarı mesajı verilecektir.)

Tek Boyutlu Dizi Uygulama

```
import java.util.Scanner;
public class Yil {
    public static void main(String[] args) {
        String Ay[]={ "", "Ocak", "Şubat", "Mart", "Nisan", "Mayıs", "Haziran",
            "Temmuz", "Ağustos", "Eylül", "Ekim", "Kasım", "Aralık" };
        Scanner tara=new Scanner(System.in);
        int y;
        System.out.println("Ay değerini giriniz");
        y=tara.nextInt();
        if (y>0 && y<=12)
            System.out.print(y+" . ay..:" + Ay[y]);
        else
            System.out.print("1 ile 12 arasında bir rakam giriniz");
    }
}
```

Tek Boyutlu Dizi Uygulama

Dışarıdan girilen N elemanlı bir dizinin tüm elemanlarını toplayan programı yazınız.

```
import java.util.Scanner;
public class Toplam {
    public static void main(String[] args) {
        Scanner tara=new Scanner(System.in);
        int N;
        int Toplam=0;
        System.out.println("Dizi boyutunu giriniz");
        N=tara.nextInt();
        int A[]=new int[N];
        for (int i=0; i<N; i++) {
            System.out.println("Dizinin " + i + ". elemanı");
            A[i]=tara.nextInt();
            Toplam=Toplam+A[i];
        }
        System.out.print("Toplam="+Toplam);
    }
}
```


for-each Benzeri Döngü Yapısı ile Dizi Elemanlarına Erişmek

- Bir dizinin elemanlarına sırayla erişmek için;
- **for(tip değişken_adı: dizi_adı)** komutu kullanılabilir. Bu komut yapısı ile dizideki her eleman sırayla döngü içinde belirtilen değişkene atanacaktır.

Örnek

1'den 10'a kadar olan sayıları A dizisine aktaran ve dizi elemanlarının toplamını ekranda gösteren programı yazınız.

Klasik for döngüsü ile bu sorunun çözümü.

```
int A[ ]={1,2,3,4,5,6,7,8,9,10};  
int top=0;  
for(int i=0; i<10; i++) top+=A[i];
```

***Ancak for döngüsünün, for-each stilinde kullanımı ile de aynı program kodlanabilir.

for-each Benzeri Döngü Yapısı ile Dizi Elemanlarına Erişmek

- Örnek 1'den 10'a kadar olan sayıları A dizisine aktaran ve dizi elemanlarının toplamını ekranda gösteren programı yazınız.

```
public class Dizi8 {  
    public static void main(String[] args) {  
        int A[]={1,2,3,4,5,6,7,8,9,10};  
        int top=0;  
        for(int x:A) {  
            System.out.println("Dizi elemanları: " + x);  
            top+=x;  
        }  
        System.out.println("Toplam: "+ top);  
    }  
}
```



Ekran Çıktısı:

Dizi elemanları: 1
Dizi elemanları: 2
Dizi elemanları: 3
Dizi elemanları: 4
Dizi elemanları: 5
Dizi elemanları: 6
Dizi elemanları: 7
Dizi elemanları: 8
Dizi elemanları: 9
Dizi elemanları: 10
Toplam: 55

Tek Boyutlu Dizi Uygulama

- Örnek 12 odalı (0'dan 11'e kadar) bir otelde, hangi odanın boş hangi odanın dolu olduğunu gösteren programı yazınız.

```
public class Otel {  
    public static void main(String[] args) {  
        String []A=new String [12];  
        A[0]="Ali Akça";  
        A[1]="Bülent Yalçın";  
        A[2]="Ayşe Koç";  
        A[3]="Aslı Gül";  
        A[7]="Zehra Ak";  
        A[A.length-1]="Otel Müdür";  
        for(int j=0;j<A.length;j++)  
            if(A[j]!=null)  
                System.out.println("Oda No "+j+": "+A[j]);  
            else  
                System.out.println("Oda No "+j+": "+"Boş");  
    }  
}
```

Çok Boyutlu Dizi Uygulama

Örnek Aşağıdaki A matrisini oluşturup ekranda gösteren programı yazınız.

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

Çok Boyutlu Dizi Uygulama

```
public class Matris {  
    public static void main(String[] args) {  
        int t,i;  
        int A[][]=new int [3][4];  
        for(t=0; t<3; ++t){  
            for(i=0;i<4;++i){  
                A[t][i]=(t*4)+i+1;  
                System.out.print(A[t][i]+"\\t");  
            }  
            System.out.println();  
        }  
    }  
}
```

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12

Çok Boyutlu Dizi Uygulama

Aşağıdaki A matrisini oluşturup ekranda gösteren programı yazınız.

0	1	2	3	4
0	1	2	3	4
0	1	2	3	4

Çok Boyutlu Dizi Uygulama

```
public class Matris2 {  
    public static void main(String[] args) {  
        int j,i;  
        int A[][]=new int [3][5];  
        for(i=0;i<3;++i){  
            for(j=0;j<5;++j){  
                A[i][j]=j;  
                System.out.print(A[i][j]+"\\t");  
            }  
            System.out.println();  
        }  
    }  
}
```

0	1	2	3	4
0	1	2	3	4
0	1	2	3	4

İki Boyutlu Dizi Uygulama

Örnek

Aşağıdaki birim matrisi oluşturup ekranda gösteren programı yazınız.

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

İki Boyutlu Dizi Uygulama

```
public class Matris3 {  
    public static void main(String[] args) {  
        int j,i;  
        int A[][]=new int [5][5];  
        for(i=0;i<5;++i){  
            for(j=0;j<5;++j){  
                if (i==j)  
                    A[i][j]=1;  
                else  
                    A[i][j]=0;  
                System.out.print(A[i][j]+" ");  
            }  
            System.out.println();  
        }  
    }  
}
```

İki Boyutlu Dizi Uygulama

Örnek

Aşağıdaki birim matrisi oluşturup ekranda gösteren programı yazınız.

0	1	1	1	1
-1	0	1	1	1
-1	-1	0	1	1
-1	-1	-1	0	1
-1	-1	-1	-1	0

İki Boyutlu Dizi Uygulama

```
public class Matris4 {  
    public static void main(String[] args) {  
        int j,i;  
        int A[][]=new int [5][5];  
        for(i=0;i<5;++i){  
            for(j=0;j<5;++j){  
                if (i==j)  
                    A[i][j]=0;  
                else if (i>j)  
                    A[i][j]=-1;  
                else  
                    A[i][j]=1;  
                System.out.print(A[i][j]+"\\t");  
            }  
            System.out.println();  
        }  
    }  
}
```

İki Boyutlu Dizi Uygulama

A[2][3] matrisinin satırları ile sütunlarını yer değiştiren programı yazınız.
(Matematikte bu işlem transpose olarak adlandırılır ve A matrisinin transposesi A^T ile gösterilir.)

A matrisi

1	2	3
4	5	6

B matrisi

1	4
2	5
3	6

İki Boyutlu Dizi Uygulama

A[2][3] matrisinin satırları ile sütunlarını yer değiştiren programı yazınız.
(Matematikte bu işlem transpose olarak adlandırılır ve A matrisinin transposesi A^T ile gösterilir.)

A matrisi

1	2	3
4	5	6

B matrisi

1	4
2	5
3	6

```
public class Matris5 {  
    public static void main(String[] args) {  
        int j,i;  
        int A[ ][ ]={{1,2,3},{4,5,6}};  
        int B[ ][ ]=new int [3][2];  
        for(i=0;i<2;++i){  
            for(j=0;j<3;++j){  
                B[j][i]=A[i][j];  
            }  
        }  
        for(i=0;i<3;++i){  
            for(j=0;j<2;++j){  
                System.out.print(B[i][j]+"\\t");  
            }  
            System.out.println();  
        }  
    }  
}
```

Kaynaklar

- Java ve Java Teknolojileri, *Tevfik KIZILÖREN* – Kodlab Yayınları
- <https://medium.com/t%C3%BCrkiye/stack-ve-heap-kavram%C4%B1-59adcb29d454https://medium.com/t%C3%BCrkiye/stack-ve-heap-kavram%C4%B1-59adcb29d454>
- http://javayaz.com/?page_id=84
- Dr Öğr. Üyesi Aysun Zehra ALTIKARDEŞ- Nesne Yönelimli Programlama Notları
- Dr. Öğr. Üyesi Erbil AKBAY – Nesne Tabanlı Programlama 2 Ders Notları