

İleri Programlama

Diziler

Hüseyin Ahmetođlu

Diziler

- ▶ 30 öğrencilik bir sınıf için notların ortalamasını bulmak istediğinizi varsayalım, kesinlikle 30 değişken oluşturmak istemezsiniz. Bunun yerine, 30 öğeli tek bir değişken kullanabilirsiniz.
- ▶ Bir dizi, bir çift köşeli parantez [] ile tanımlanan aynı türdeki öğelerin sıralı bir koleksiyonudur.
- ▶ Dizi bir ad ve türle bildirilir. Dizinin tüm elemanları aynı türe aittir.

Diziler

Bellekte sürekli yer kaplayan aynı türden verilerin oluşturduğu kümeye dizi adı verilir. Tanımdan da anlaşılacağı üzere bir dizinin dizi olabilmesi için:

1. Bellekte sürekli yer kaplaması
2. Aynı tür elemanlardan oluşmuş olması gerekir.

Dizilerin her elemanı ayrı bir nesne (*object*) gibi ele alınmalıdır. Dizi tanımlarken; o dizinin adını ve dizi içerisindeki elemanlara erişmek amacıyla onun kaçınıcı eleman olduğunu belirten indisini yazmak gerekir.

Java dilinde dizi tanımlamak için:

```
Tip [ ] dizi adı = new Tip [Eleman Sayısı];
```

veya

```
Tip dizi adı [ ] = new Tip [Eleman Sayısı];
```

yapıları kullanılabilir.

Diziler

Java dilinde dizilerin ilk elemanı, 0. indisli elemanıdır. Son elemanı da dizi tanımında belirtilen eleman sayısının 1 eksiğidir.

A[7]	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
	5	7	9	23	45	12	04

İçerisinde tamsayıları sakladığımız 7 elemanlı bir A dizisini aşağıdaki gibi tanımlayabiliriz:

```
1. int [ ] A=new int[7];
2. int A[ ];
   A=new int[7];
3. int A [] = new int [7];
```

Bu şekilde tanımlanan 7 elemanlı bir A dizisinin ilk elemanı A[0], son elemanı ise A[6] indisli elemanıdır. Dizinin herhangi bir elemanına dizi adı ve köşeli parantez içerisindeki indis numarası ile birlikte erişilir. Örneğin 'A dizisinin 2. indisli elemanı' A[2] şeklinde okunabilir ve değeri bu örnek için 9'dur.

▪

Diziler

- ▶ new operatörü aracılığıyla bir dizi oluşturulduğunda, tüm öğeler varsayılan değerlerine, örneğin, int için 0, double için 0.0, boolean için false ve nesnelere için null değerleriyle başlatılır.

```
int[] marks; // Declare an int array named "marks"
             // "marks" is assigned to a special value called "null" before allocation
int marks[]; // Same as above, but the above syntax recommended
marks = new int[5]; // Allocate 5 elements via the "new" operator
// Declare and allocate a 20-element array in one statement via "new" operator
int[] factors = new int[20];
// Declare, allocate a 6-element array thru initialization
int[] numbers = {11, 22, 33, 44, 55, 66}; // size of array deduced from the number of items
```

İndex Kavramı

Dizilerde değişken ismini takiben köşeli parantez içerisinde belirtilen numaraya *indis* (*index/subscript*) adı verilir. Değişken tanımlamada olduğu gibi dizi tanımlamada da veri tipini mutlaka belirtmemiz gerekir. Java dilinde dizinin indisi köşeli parantez içerisinde belirtilir.

Dizi ve indislere isim verirken dikkat edilmesi gereken hususlar:

- İndisler negatif olamaz.
- İndisler tamsayı veya tamsayı üreten bir değer olmalıdır.
- İndis değeri en az 1 olmalıdır.
- Dizi tanımlama esnasında indis değeri verilmez (boyutu belirtilmez) ise dizinin boyutu tanımlanan eleman sayısı kadardır. Örneğin `int A[] = {1, 2, 3, 4, 5};` şeklindeki tanımlamada dizinin boyutu 5'dir.
- Dizinin ilk indis değeri 0'dır. (sıfır) Dolayısıyla n elemanlı bir dizinin n. elemanının indis değeri n-1'dir.
- Bir dizinin eleman sayısını (boyutunu) bulmak için `length` metodu kullanılır. Bu metodun döndürdüğü değer `int` tipindedir. Örneğin `int A[] = { 8, 9, 12, 4, 5, 7, 1, 0, 6 }` şeklinde tanımlanan bir dizinin boyutunu (eleman sayısını) öğrenmek için `A.length` komutunu kullanırız. Dizinin son indisli elemanı ise `A.length - 1`'dir.

Array Index

- Köşeli parantez [] içine alınmış bir index aracılığıyla bir dizinin bir ögesine başvurabilirsiniz. Java'nın dizi index'i sıfır (0) ile başlar.

```
int[] marks = new int[5]; // Declare & allocate a 5-element int array
// Assign values to the elements
marks[0] = 95;
marks[1] = 85;
marks[2] = 77;
marks[3] = 69;
marks[4] = 66;
// Retrieve elements of the array
System.out.println(marks[0]);
System.out.println(marks[3] + marks[4]);
```

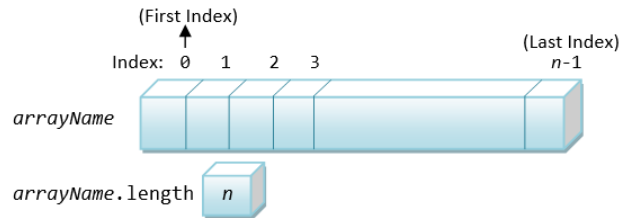
Dizi Boyutu

- ▶ Bir dizi oluşturmak için dizinin uzunluğunu (veya boyutunu) önceden bilmeniz ve buna göre tanımlamanız gerekir.
- ▶ Bir dizi oluşturulduktan sonra uzunluğu sabittir ve çalışma sırasında değiştirilemez.
- ▶ Bazen bir dizinin uzunluğunu belirlemek zordur (örneğin, kaç öğrenci?).
- ▶ Bununla birlikte, uzunluğu tahmin etmeniz ve bir üst sınır belirlemeniz gerekir.
- ▶ Uzunluğu 30 olarak ayarladığınızı (bir öğrenci sınıfı için) ve 31 öğrenci olduğunu, yeni bir dizi (uzunluk 31) atamanız, eski diziyi yeni diziyeye kopyalamanız ve eski diziyi silmeniz gerektiğini varsayalım. Başka bir deyişle, bir dizinin uzunluğu çalışma süresi sırasında dinamik olarak ayarlanamaz. Bu muhtemelen bir dizi kullanmanın en büyük dezavantajıdır. (Dinamik olarak ayarlanabilen başka yapılar da vardır.)

Dizi Boyutu

- Bir dizde index 0 ile `diziAdı.length - 1` arasındadır.
- Eğer index `[0, arrayName.length-1]` aralığının dışındaysa, Java Runtime `ArrayIndexOutOfBoundsException` adlı bir istisna hatasını verir.

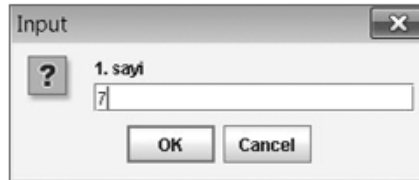
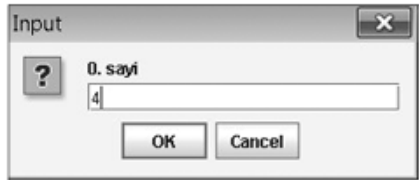
```
int[] factors = new int[5];           // Declare and allocate a 5-element int array
int numFactors = factors.length;     // numFactor is 5
```



Bir diziye değer aktarımı

```
import javax.swing.JOptionPane;
class Ornek5 {
    public static void main ( String[] args )
    {
        int[] Sayi= new int[5];
        // dizi elemanları giriliyor
        for ( int i=0; i < Sayi.length; i++ )
        {
            Sayi[i] = Integer.parseInt(JOptionPane.showInputDialog(i+ ". sayi"));
        }
        // dizi elemanları ekranda gösteriliyor
        for ( int i=0; i < Sayi.length; i++ )
        {
            System.out.println( "Sayi[ " + i + " ] = " + Sayi[ i ] );
        }
    }
}
```

Programın ekran çıktısı:



Dizi elemanları Input iletişim kutuları ile yukarıdaki gibi tek tek girilir ve girilen elemanlar ekranda/konsolda yandaki gibi görülür.

..

```
<terminated> Ornek5
Sayi[ 0 ] = 4
Sayi[ 1 ] = 7
Sayi[ 2 ] = 8
Sayi[ 3 ] = 12
Sayi[ 4 ] = 9
```

Diziler ve Döngüler

- Diziler, döngülerle iç içe çalışır. Bir dizinin tüm elemanlarını bir döngü aracılığıyla işleyebilirsiniz,

```
1  /**
2   * Find the mean and standard deviation of numbers kept in an array
3   */
4  public class MeanSDArray {
5      public static void main(String[] args) {
6          // Declare variable
7          int[] marks = {74, 43, 58, 60, 90, 64, 70};
8          int sum = 0;
9          int sumSq = 0;
10         double mean, stdDev;
11
12         // Compute sum and square-sum using loop
13         for (int i = 0; i < marks.length; ++i) {
14             sum += marks[i];
15             sumSq += marks[i] * marks[i];
16         }
17         mean = (double)sum / marks.length;
18         stdDev = Math.sqrt((double)sumSq / marks.length - mean * mean);
19
20         // Print results
21         System.out.printf("Mean is: %.2f%n", mean);
22         System.out.printf("Standard deviation is: %.2f%n", stdDev);
23     }
24 }
```

for-each

- ▶ Döngü, dizideki her öge için bir kez çalıştırılır ve ögenin değeri bildirilen değişkene kopyalanır. Daha az kod satırı gerektirir, döngü sayacını ve dizi indeksini ortadan kaldırır ve okunması daha kolaydır. Ancak, ilkel türler dizisi için (ör., int), yalnızca öğeleri okuyabilir ve dizinin içeriğini değiştiremez. Bunun nedeni, her elemanın değerinin, orijinal kopyası üzerinde çalışmak yerine döngünün değişkenine kopyalanmasıdır.

Syntax

```
for (type item : anArray) {  
    body ;  
}  
  
// type must be the same as the  
// anArray's type
```

```
int[] numbers = {8, 2, 6, 4, 3};  
int sum = 0, sumSq = 0;  
for (int number : numbers) { // for each int number in int[] numbers  
    sum += number;  
    sumSq += number * number;  
}  
System.out.println("The sum is: " + sum);  
System.out.println("The square sum is: " + sumSq);
```

```

import java.util.Scanner;
/**
 * Prompt user for the length and all the elements of an array; and print [a1, a2, ..., an]
 */
public class ReadPrintArray {
    public static void main(String[] args) {
        // Declare variables
        int numItems;
        int[] items; // Declare array name, to be allocated after numItems is known
        Scanner in = new Scanner(System.in);

        // Prompt for a non-negative integer for the number of items;
        // and read the input as "int". No input validation.
        System.out.print("Enter the number of items: ");
        numItems = in.nextInt();

        // Allocate the array
        items = new int[numItems];

        // Prompt and read the items into the "int" array, only if array length > 0
        if (items.length > 0) {
            System.out.print("Enter the value of all items (separated by space): ");
            for (int i = 0; i < items.length; ++i) {
                items[i] = in.nextInt();
            }
        }

        // Print array contents, need to handle first item and subsequent items differently
        System.out.print("The values are: []");
        for (int i = 0; i < items.length; ++i) {
            if (i == 0) {
                // Print the first item without a leading commas
                System.out.print(items[0]);
            } else {
                // Print the subsequent items with a leading commas
                System.out.print(", " + items[i]);
            }
        }
        System.out.println("]");
        in.close();
    }
}

```

```

Enter the number of items: 5
Enter the value of all items (separated by space): 7 9 1 6 2
The values are: [7, 9, 1, 6, 2]

```

Arrays.toString()

```
import java.util.Arrays; // Needed to use Arrays.toString()
/**
 * Use Arrays.toString() to print an array in the form of [a1, a2, ..., an]
 */
public class TestArrayToString {
    public static void main(String[] args) {
        // Declare and allocate test arrays
        int[] a1 = {6 ,1, 3, 4, 5}; // Allocate via initialization
        int[] a2 = {}; // Empty array with length = 0
        double[] a3 = new double[1]; // One-Element array, init to 0.0

        System.out.println(Arrays.toString(a1)); //[6, 1, 3, 4, 5]
        System.out.println(Arrays.toString(a2)); [[]]
        System.out.println(Arrays.toString(a3)); //[0.0]
        a3[0] = 2.2;
        System.out.println(Arrays.toString(a3)); //[2.2]
    }
}
```

Örnek 6.10: Klavyeden girilen para miktarını şu anda kullanılmakta olan para kupürlerine (100-50-20-10-5-1) göre en az sayıda kupür ile ödemek için bir veznedara yardımcı olacak programı yazınız.

```
import java.util.Scanner;
class Kupur {
public static void main (String args[]) {
int kupur[] = new int[6];
int sayi;
Scanner tara =new Scanner(System.in) ;
int para;
kupur[0] = 100;
kupur[1] = 50;
kupur[2] = 20;
kupur[3] = 10;
kupur[4] = 5;
kupur[5] = 1;
```

```
System.out.println("Para değerini giriniz");
para=tara. nextInt();
for(int i=0; i<6;i++) {
sayi=para/kupur[i];
if (sayi!=0)
{
System.out.println(sayi+"adet"+kupur[i]);
}
para = para - sayi * kupur[i];
}
}
```

Örnek 6.13: a dizisinin 2. elemanından itibaren 5 elemanını b dizisine kopyalayan programı yazınız.

Çözüm:

a dizisi	<code>char[] a = {'a','b','c','d','e','f','g','h','i','j'};</code>										
	<table border="1"><tr><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td><td>f</td><td>g</td><td>h</td><td>i</td><td>j</td></tr></table> <code>a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7] a[8] a[9]</code>	a	b	c	d	e	f	g	h	i	j
a	b	c	d	e	f	g	h	i	j		
Kopyalanıyor	<code>System.arraycopy(a, 2, b, 0, 5);</code>										
b dizisi	<code>'c','d','e','f','g'</code>										
	<table border="1"><tr><td>c</td><td>d</td><td>e</td><td>f</td><td>g</td></tr></table> <code>a[0] a[1] a[2] a[3] a[4]</code>	c	d	e	f	g					
c	d	e	f	g							

```
class DiziKopyala {  
    public static void main(String[] args) {  
        char[] a = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j' };  
        char[] b = new char[5];  
        System.arraycopy(a, 2, b, 0, 5);  
        System.out.println(b);  
    }  
}
```

Programın ekran çıktısı yandaki gibidir:

<terminated> DiziKopyala
cdefg

Örnek 6.15: { 50, 20, 45, 82, 25, 63 } elemanlarından oluşan bir A dizisinin elemanlarını küçükten büyüğe doğru sıralayan ve bu dizinin en küçük ve en büyük elemanını ekranda gösteren programı yazınız.

Çözüm:

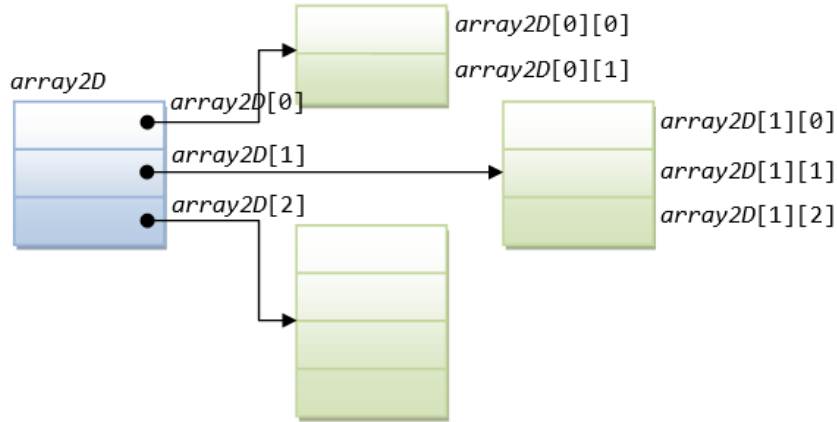
```
import static java.util.Arrays.sort;

public class DiziElemanlarınıSırala {
    public static void main(String[] args) {
        int A[] = { 50, 20, 45, 82, 25, 63 };
        System.out.print("Dizinin sıralı hali...: ");
        sort(A);
        for (int i = 0; i < A.length; i++) {
            System.out.print("  " + A[i]);
        }
        System.out.print("\r\nEn küçük eleman= " + A[0]);
        System.out.print("\r\nEn büyük eleman= " + A[A.length - 1]);
    }
}
```

Programın ekran çıktısı:

```
<terminated> DiziElemanlarınıSırala [Java Application]
Dizinin sıralı hali...: 20 25 45 50 63 82
En küçük eleman= 20
En büyük eleman= 82
```

Çok Boyutlu Dizi



- Matrisler ile işlem yaparken ya matrisin satırlarında sütunlar içerisinde, ya da matrisin sütunlarında satırlar içerisinde dolaşırız.
- Matematiksel olarak A_{ij} şeklinde tanımlanan A matrisi, bilgisayar ortamında $A[i][j]$ şeklinde gösterilir. Böyle bir matrisin eleman sayısı $i*j$ 'dir.

$$A(i,j) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1j} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2j} \\ \dots & \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & a_{i3} & \dots & a_{ij} \end{bmatrix}$$

Satır sayısı _____
Sütun sayısı _____

Çok Boyutlu Dizi

```
int grid[][] = new int[12][8]; // a 12x8 grid of int
grid[0][0] = 8;
grid[1][1] = 5;
System.out.println(grid.length); // 12
System.out.println(grid[0].length); // 8
System.out.println(grid[11].length); // 8
```

```
public class Array2DTest {
    public static void main(String[] args) {
        int[][] grid = new int[12][8]; // A 12x8 grid, in [row][col] or [y][x]
        int numRows = grid.length; // 12
        int numCols = grid[0].length; // 8

        // Fill in grid
        for (int row = 0; row < numRows; ++row) {
            for (int col = 0; col < numCols; ++col) {
                grid[row][col] = row*numCols + col + 1;
            }
        }

        // Print grid
        for (int row = 0; row < numRows; ++row) {
            for (int col = 0; col < numCols; ++col) {
                System.out.printf("%3d", grid[row][col]);
            }
            System.out.println();
        }
    }
}
```

Çok Boyutlu Dizi

```
1 public class Array2DWithDifferentLength {
2     public static void main(String[] args) {
3         int[][] grid = {
4             {1, 2},
5             {3, 4, 5},
6             {6, 7, 8, 9}
7         };
8
9         // Print grid
10        for (int y = 0; y < grid.length; ++y) {
11            for (int x = 0; x < grid[y].length; ++x) {
12                System.out.printf("%2d", grid[y][x]);
13            }
14            System.out.println();
15        }
16
17        // Another 2D array
18        int[][] grid1 = new int[3][];
19        grid1[0] = new int[2];
20        grid1[1] = new int[3];
21        grid1[2] = new int[4];
22
23        // Print grid - all elements init to 0
24        for (int y = 0; y < grid1.length; ++y) {
25            for (int x = 0; x < grid1[y].length; ++x) {
26                System.out.printf("%2d", grid1[y][x]);
27            }
28            System.out.println();
29        }
30    }
31 }
```

Örnek6.23: Yandaki Amatrisini oluşturup ekranda gösteren programı yazınız.

Çözüm: Bu A matrisini $A[i][j]$ olarak tanımlarsak, i satır elemanlarını, j sütun elemanlarını göstermektedir. Birim matris örneğinin tersine $i=j$ ise matris 0, değilse 1 ya da -1 değerlerini almaktadır. 1 ya da -1 olma durumuna dikkat edersek; A matrisinin satır elemanları, sütun elemanlarından büyük olduğunda ($i > j$ durumu) -1, diğer durumda 1 değerini almaktadır.

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

```
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();
}
}}
```

Programın ekran çıktısı aşağıdaki gibidir:

General Output				
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

Örnek 6.24: Bir matrisin satırları ile sütunlarını yer değiştiren programı yazınız. (Matematikte bu işlem transpose olarak adlandırılır ve A matrisinin transpozesi A^T ile gösterilir.)

Çözüm: $A[2][3]$ şeklindeki bir matrisin transpozesi $B[3][2]$ şeklinde görülür. Örneğin;

A matrisi

1	2	3
4	5	6

şeklinde ise

B matrisi (*A matrisinin transpozesi*)

1	4
2	5
3	6

şeklinde dir.

```
class Transpose {
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]; //A matrisinin transpozesi
}
}
for(i=0; i < 3; ++i) {
for(j=0; j < 2; ++j) {
System.out.print(B[i][j] + " ");
}
System.out.println();
}
}}
```

Örnek 6.27: $M \times N$ elemanlı bir A matrisinin içindeki tek sayıları, tek boyutlu B dizisine aktaran programı yazınız.

Çözüm:

- Örnek: $A_{2 \times 3} = \begin{bmatrix} 2 & 3 & 4 \\ 9 & 8 & 1 \end{bmatrix}$ şeklindeki matristen $B = \{ 0, 3, 0, 9, 0, 1 \}$ dizisi elde edilecek.
- Bir sayının tek mi çift mi olduğunu anlamanın en kolay yolu, sayının 2'ye bölünmeden kalan (sayı % 2) sıfırsa sayı çift, değilse tektir. B dizisi, A dizisinin tek sayılarından ve çift sayıları yerine de sıfır (0) değerlerinden oluşacaktır.

Java programlama dilinde kodlanması:

```
import java.util.Scanner;
class Matris26 {
    public static void main(String args[] ) {
        Scanner tara = new Scanner(System.in);
        System.out.println("A Matrisinin satır değerini giriniz");
        int M = tara.nextInt();
        System.out.println("A Matrisinin sutun değerini giriniz");
        int N = tara.nextInt();
        int A[][] = new int[M][N];
        int B[] = new int[M * N];
        int s = 0;
        // A matrisinin elemanları giriliyor
        for (int i = 0; i < M; i++) {
            for (int j = 0; j < N; j++) {
                System.out.print(" A [" + i + ", " + j + "]...: ");
                A[i][j] = tara.nextInt();
            }
            System.out.println();
        }
        // A matrisinden B dizisi oluşturuluyor
        for (int i = 0; i < M; i++) {
            for (int j = 0; j < N; j++) {
                if (A[i][j] % 2 != 0)
                    B[s] = A[i][j];
                s = s + 1;
            }
        }
        System.out.println("B dizisinin elemanları");
        for (int i = 0; i < s; i++) {
            System.out.print(B[i] + "\t");
        }
    }
}
```

KAYNAKLAR

- ▶ Programming Notes. (2021, March 11). Retrieved from <https://www3.ntu.edu.sg/home/ehchua/programming/index.html>
- ▶ Çobanoğlu B. (2020) Java ile Programlama ve Veri Yapıları. İstanbul Pusula Yayıncılık. 978-605-2359-84-6