

Javada Diziler

Java da diziler nesnedir; içerisinde belirli sayıda eleman bulunur. Eğer bu sayı sıfır ise, dizi boş demektir. Dizinin içerisindeki elemanlara eksi olmayan bir tam sayı ile ifade edilen dizi erişim indisi ile erişilir. Bir dizide n tane eleman varsa dizinin uzunluğu da n kadardır; ilk elemanın indisi/konumu 0'dan başlar, son elemanı ise n-1'dir.

Dizi içerisindeki elemanlar aynı türden olmak zorundadır. Eğer dizi içerisindeki elemanların türü double ise, bu dizinin türü için double denilir. Bu double tipinde olan diziye String tipinde bir nesne atanması denenirse hata ile karşılaşılır. Diziler temel veya herhangi bir sınıf tipinde olabilir...

Dizi Türündeki Referanslar

Dizi türündeki referanslar, dizi nesnelere bağlanmaktadır. Dizi referansları tanımlamak bu dizinin hemen kullanılacağı anlamına gelmez...

```
double[] dd ; // double tipindeki dizi
double dd[] ; // double tipindeki dizi
float [] fd ; // float tipindeki dizi
Object[] ao ; // Object tipindeki dizi
```

Dizi nesnelerini oluşturmak için **new** anahtar sözcüğü kullanılması gereklidir.

```
double[] d = new double[20]; // 20 elemanlı double tipindeki dizi
double dd[] = new double[20]; // 20 elemanlı double tipindeki dizi
float []fd = new float [14]; // 14 elemanlı float tipindeki dizi
Object[]ao = new Object[17]; // 17 elemanlı Object tipindeki dizi
String[] s = new String[25]; // 25 elemanlı String tipindeki dizi
```

Dizi Boyunun Değiştirilmesi

Dizi boyutu birkez verildi mi, artık değiştirilemezler!

:

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

liste = new int[15] ; // yeni bir dizi nesnesi bağlandı
```

Yukarıda dizi boyutlarının büyütüldüğünü sanmayın; burada, yalnızca yeni bir dizi nesnesi daha oluşturulmaktadır. liste dizi referansının daha evvelden bağlanmış olduğu dizi nesnesi (new int[5]), çöp toplama işlemi sırasında çöp toplayıcısı tarafından bellekten silinecektir.

Dizi Elemanlarına Erişim

Java dilinde dizi kullanımı diğer dillere nazaran daha az risklidir; anlamı, eğer tanımladığımız dizinin sınırları aşılsak, çalışma-anında (runtime) hata ile karşılaşacağımızdır. Örneğin 20 elemanlı bir double dizisi tanımlanmış olsun. Eğer bu dizinin 78. elemanına ulaşmak istenirse (- ki böyle bir indisli eleman yok), olanlar olur ve çalışma-anında hata alınır; böylesi hatanın (ArrayIndexOutOfBoundsException Exception) çalışma-anında alınması güvenlik açısından güzel bir olaydır. Böylece dizi için ayrılmış bellek alanından dışarı çıkılıp başka verilere müdahale edilmesi engellenmiş olunur.

Örnek

```
public class DiziElemanlariGosterimBir {

    public static void main(String args[]) {
        double[] d = {2.1, 3.4, 4.6, 1.1, 0.11};
        String[] s = {"defter", "kalem", "sarman", "tekir", "boncuk"};
        ;
        // double tipindeki dizimizi ekrana yazdırıyoruz
        for (int i = ; i < d.length; i++) {
            System.out.println("d[" + i + "] = " + d[i]);
        }
        // System.out.println("d["+78+"] = " + d[78] ); // Hata !
        System.out.println("-----");
        // String tipindeki dizimizi ekrana yazdırıyoruz
        for (int x = ; x < s.length; x++) {
            System.out.println("s[" + x + "] = " + s[x]);
        }
        // System.out.println("s["+78+"]=" + s[78] ); // Hata !
    }
}
```

Length ifadesiyle bir dizinin içerisindeki eleman sayısı öğrenilir. Bu örnekte iki adet dizi tanımlandı:double ve String türündeki dizilerin içerisine 5'er adet eleman yerleştirildi ve sonradan bunları for döngüsü ile ekrana yazdırıldı. i < d.length ifadesine dikkat edilirse, döngü sayacın 4'e kadar artmaktadır; ancak, döngü sayacının 0'dan başladığı unutulmamalıdır.

Eğer 5 elemana sahip olan dizinin 78. elemanına erişilmeye kalkılırsa, derleme anında (compile-time) bir hata ile karşılaşılmaz; ancak, uygulama yürütüldüğü zaman; yani, çalışma-anında (runtime) hata ile karşılaşılır. Uygulamanın sonucu aşağıdaki gibi olur:

```
d[] = 2.1
d[1] = 3.4
d[2] = 4.6
d[3] = 1.1
d[4] = 0.11
```

```
s[] = defter
s[1] = kalem
s[2] = sarman
s[3] = tekir
s[4] = boncuk
```

Bir önceki uygulamanın çalışma anına hata vermesi istenmiyorsa, yorum satırı olan yerler açılması ve uygulamanın baştan derlenip çalıştırması gerekmektedir. Aynı örnek daha değişik bir şekilde ifade edilebilir:

DiziElemanlariGosterimIki.java

```
public class DiziElemanlariGosterimIki {

    double[] d;
    String[] s;

    public DiziElemanlariGosterimIki() { // double tipindeki diziye eleman atanıyor
        d = new double[5];
        d[] = 2.1;
        d[1] = 3.4;
        d[2] = 4.6;
        d[3] = 1.1;
        d[4] = 0.11;
        // d[5] = 0.56 ; // Hata !
        // String tipindeki diziye eleman atanıyor
        s = new String[5];
        s[] = new String("defter");
        s[1] = new String("kalem");
        s[2] = new String("sarman");
        s[3] = new String("tekir");
        s[4] = new String("boncuk");
        // s[5] = new String("duman"); // Hata !
    }

    public void ekranaBas() { // double tipindeki diziyi ekrana yazdırıyoruz
        for (int i = ; i < d.length; i++) {
            System.out.println("d[" + i + "] = " + d[i]);
        }
        System.out.println("-----"); // String dizi ekrana yazdırılıyor
        for (int x = ; x < s.length; x++) {
            System.out.println("s[" + x + "] = " + s[x]);
        }
    }

    public static void main(String args[]) {
        DiziElemanlariGosterimIki deg = new DiziElemanlariGosterimIki();
        deg.ekranaBas();
    }
}
```

```
}
```

Bu örnekte 5 elemanlı dizilere 6. eleman eklenmeye çalışıldığında, derleme anında (compile-time) herhangi bir hata ile karşılaşmayız. Hata ile karşılaşacağımız yer çalışma anındadır. Çünkü bu tür hatalar çalışma anında kontrol edilir. Yalnız çalışma anında hata oluşturabilecek olan satırlar kapatıldığı için şu an için herhangi bir tehlike yoktur; ancak, çalışma anında bu hatalar ile tanışmak isterseniz, bu satırların başında `"/"` yorum ekini kaldırmanız yeterli olacaktır. Uygulamanın sonucu aşağıdaki gibi olacaktır:

```
d[0] = 2.1
d[1] = 3.4
d[2] = 4.6
d[3] = 1.1
d[4] = 0.11
-----
s[0] = defter
s[1] = kalem
s[2] = sarman
s[3] = tekir
s[4] = boncuk
```

Dizilerin Elemanlarını Sıralama

Dizi elemanlarını büyükten küçüğe doğru sıralatmak için `java.util` paketini altındaki `Arrays` sınıfı kullanılabilir. Bu sınıfın statik `sort()` yordamı sayesinde dizilerin içerisindeki elemanlar sıralanabilir:

Örnek

DiziSiralama.java

```
import java.util.*; // java.util kütüphanesini kullanmak için

public class DiziSiralama {

    public static void ekranaBas(double[] d) {
        for (int i = 0; i < d.length; i++) {
            System.out.println("d[" + i + "] = " + d[i]);
        }
    }

    public static void main(String args[]) {
        double d[] = new double[9];
        d[0] = 2.45;
        d[1] = 3.45;
        d[2] = 4.78;
        d[3] = 1.45;
        d[4] = 15.12;
        d[5] = 1;
```

```

        d[6] = 9;
        d[7] = 15.32;
        d[8] = 78.17;
        System.out.println("Karisik sirada");
        ekranaBas(d);
        Arrays.sort(d);
        System.out.println("Siralanmis Sirada");
        ekranaBas(d);
    }
}

```

Uygulama sonucu aşağıdaki gibi olur:

Karisik sirada

```

d[] = 2.45
d[1] = 3.45
d[2] = 4.78
d[3] = 1.45
d[4] = 15.12
d[5] = 1.0
d[6] = 9.0
d[7] = 15.32
d[8] = 78.17

```

Siralanmis Sirada

```

d[] = 1.0
d[1] = 1.45
d[2] = 2.45
d[3] = 3.45
d[4] = 4.78
d[5] = 9.0
d[6] = 15.12
d[7] = 15.32
d[8] = 78.17

```

Dizilerin Dizilere Kopyalanması

Bir dizi tümünden diğer bir diziye kopyalanabilir:Bunu bir örnekle açıklayalım.

Örnek:

DiziSiralama.java

```
import java.util.*; // java.util kütüphanesini kullanmak için

public class DiziSiralama {

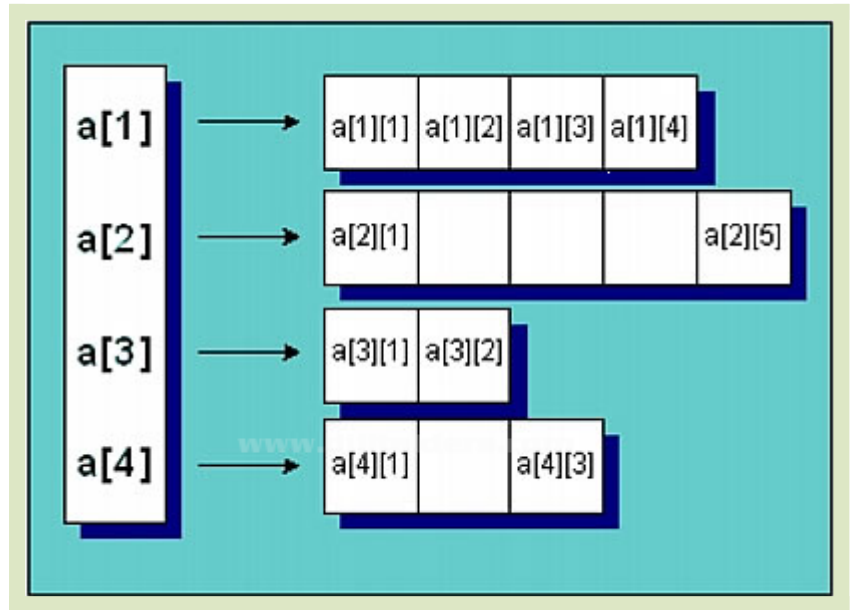
    public static void ekranaBas(double[] d) {
        for (int i = 0; i < d.length; i++) {
            System.out.println("d[" + i + "] = " + d[i]);
        }
    }

    public static void main(String args[]) {
        double d[] = new double[9];
        d[0] = 2.45;
        d[1] = 3.45;
        d[2] = 4.78;
        d[3] = 1.45;
        d[4] = 15.12;
        d[5] = 1;
        d[6] = 9;
        d[7] = 15.32;
        d[8] = 78.17;
        System.out.println("Karisik sirada");
        ekranaBas(d);
        Arrays.sort(d);
        System.out.println("Siralanmis Sirada");
        ekranaBas(d);
    }
}
```

System sınıfının statik yordamı olan **arraycopy()** sayesinde dizi1 dizi2'ye kopyalandı. Sonuç aşağıdaki gibi olur:

```
dizi2[0] = 1
dizi2[1] = 2
dizi2[2] = 3
dizi2[3] = 4
dizi2[4] = 40
dizi2[5] = 30
dizi2[6] = 20
dizi2[7] = 10
```

Çok Boyutlu Diziler



Çok boyutlu diziler, Java'da diğer programlama dillerinden farklıdır. Sonuçta dizinin tek türde olması gerekir. Yani, dizi içerisinde diziler tanımlayabilirsiniz.

```
int[ ] t1 = {{ 1, 2, 3 }, { 4, 5, 6 }};
```

Yukarıda ifade edildiği gibi iki boyutlu temel türden oluşmuş çok boyutlu dizi oluşturulabilir. Çok boyutlu dizileri oluşturmanın diğer bir yolu ise,

```
int [ ]t1 = new int [3][4] ;
int [ ] t1 = new int [4] ; // ! Hata !
```

Çok boyutlu dizileri bir uygulama üzerinde incelersek;

Örnek:

CokBoyutluDizilerOrnekBir.java

```
public class CokBoyutluDizilerOrnekBir {

    public static void main(String args[]) {
        int ikiboyutlu[][] = new int[3][4];
        ikiboyutlu[][] = 45;
        ikiboyutlu[][1] = 83;
```

```

        ikiboyutlu[][2] = 11;
        ikiboyutlu[][3] = 18;
        ikiboyutlu[1][] = 17;
        ikiboyutlu[1][1] = 56;
        ikiboyutlu[1][2] = 26;
        ikiboyutlu[1][3] = 79;
        ikiboyutlu[2][] = 3;
        ikiboyutlu[2][1] = 93;
        ikiboyutlu[2][2] = 43;
        ikiboyutlu[2][3] = 12;
// ekrana yazdırıyoruz
        for (int i = ; i < ikiboyutlu.length; i++) {
            for (int j = ; j < ikiboyutlu[i].length; j++) {
                System.out.println(" ikiboyutlu[" + i + "][" + j + "] = " +
ikiboyutlu[i][j]);
            }
        }
    }
}

```

Verilen örnekte int türünde 3'e 4'lük (3x4) çok boyutlu dizi oluşturuldu; bu diziyi 3'e 4'lük bir matris gibi de düşünülebilir. Uygulama sonucu aşağıdaki gibi olur:

```

ikiboyutlu[][] =45
ikiboyutlu[][1] =83
ikiboyutlu[][2] =11
ikiboyutlu[][3] =18
ikiboyutlu[1][] =17
ikiboyutlu[1][1] =56
ikiboyutlu[1][2] =26
ikiboyutlu[1][3] =79
ikiboyutlu[2][] =3
ikiboyutlu[2][1] =93
ikiboyutlu[2][2] =43
ikiboyutlu[2][3] =12

```

Uygulama sonucu matris gibi düşünülürse aşağıdaki gibi olur:

```

45 83 11 18

17 56 26 79

3 93 43 12

```




Kaynak: Burak Taş