

DİZİLER

Diziler (Arrays), aynı türden verilerin daha kolay gruplandırıldığı ve yönetildiği yapılardır. Bir dizi içindeki veriler, 0'dan başlayıp numaralandırılan **indis değerleri** ile kod yapısında çalıştırılırlar.

```
// Dizilerin Kullanımı...
import java.util.Arrays;
import java.util.Scanner;

public class diziler1 {
    public static void main(String[] args) {

        /*
        Dizi Tanımlaması:
        5 Öğrenci Kapasiteli Bir Kursun Öğrenci Sistemi      Dizi
        Elemanları sıra ile 0 1 2 şeklinde index te yer alır.

        1. elemanın Dizi İndis Numarası: 0
        2. elemanın Dizi İndis Numarası: 1
        3. elemanın Dizi İndis Numarası: 2 olarak tanımlıdır.
        */
        //veriTipi [] diziAdı = new veriTipi [elemanSayısı];
        int [] ogrNoDizisi = new int[3];

        // Dizi İçerisine Eleman Tanımlaması:
        ogrNoDizisi[0]=107;
        ogrNoDizisi[1]=210;
        ogrNoDizisi[2]=78;

        // For Döngüsü İle Dizi İçerisindeki Elemanlara Erişme:

        for(int i=0;i<ogrNoDizisi.length;i++) {
            System.out.println("Dizinin " + i + ".Değeri: " +
ogrNoDizisi[i]);
        }

        // Dizi verilerini küçükten büyüğe sıralamak için
        "Arrays.sort(diziAdi);" komutunu kullanacağız.

        // Programımızda "Arrays.sort" komutunu kullanmak için
        "java.util" kütüphanesinden "Arrays" sınıfını tanımlayalım.
        Arrays.sort(ogrNoDizisi);
        // Dizi Elemanlarını Ekrana Yazdırma:
        for(int f=0;f<ogrNoDizisi.length;f++){
            System.out.print(ogrNoDizisi[f]+"\\n");    }
```

```

// 2 Dizinin birbiri ile karşılaştırılması için EQUALS () METODU
//ogrNoDizisi ni küçükten büyüğe sıraladık. Ona göre karşılaştırma
yapar.

    int [] kursaGelen = {78, 107,210};
    if(Arrays.equals(ogrNoDizisi,kursaGelen)==true)
        System.out.println("Dizideki veriler aynıdır.");
    else
        System.out.println("Dizideki veriler farklıdır.");


// DİZİDE BİR VERİ ARAMA
// kullanıcıdan veri alacağımız için Scanner sınıfı kullanırız.
Java.util paketinden Scanner sınıfını import ederiz.
System.out.print("Dizi İçerisinde Aranacak Değer: ");
    Scanner scan = new Scanner (System.in);
    int arananDeger=scan.nextInt();
    int indis = Arrays.binarySearch(ogrNoDizisi,arananDeger);
    System.out.print("-----\n- Dizi Elemanları:");
    for(int i=0;i<ogrNoDizisi.length;i++){
        System.out.print("[ "+ogrNoDizisi[i]+" ]");
    }
    System.out.println("\n-----");
    if(indis<0)
        System.out.println("Arama Yaptığınız Değer Bulunamadı.");
    else
        System.out.println("Arama Yaptığınız "+arananDeger+"
Değeri "+indis+" İndisinde Bulunuyor.");
    }
}

```

ÇOK BOYUTLU DİZİLER

// Dizideki değerleri döngü içerisinde kullanmak.

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

        String[] ogrenciler= {"Mücahit",
"Talha", "Alperen", "Yağız", "Bera"};
        for (String a: ogrenciler) {
            System.out.println(a);
        }
    }
}
```

// İki boyutlu dizileri iç içe döngülerle yazdırmak...

```
public class Main2 {
    public static void main(String[] args) {

String[] [] ogrenciler = new String [3] [2];

        ogrenciler[0][0] = "Mücahit";
        ogrenciler[0][1] = "Matematik";
        ogrenciler[1][0] = "Talha";
        ogrenciler[1][1] = "Tarih";
        ogrenciler[2][0] = "Alperen";
        ogrenciler[2][1] = "Yabancı Dil";

        for (int i=0; i<3; i++) {

            for (int j=0; j<2; j++) {
                System.out.print(ogrenciler[i][j]);
            }

            System.out.println ("");
        }
    }
}
```

// İki boyutlu dizileri iç içe döngülerle yazdırmak...

```
public class Main3 {

    public static void main(String[] args) {
        String[] [] ogrenciler = {"Mücahit",
"Talha", "Alperen", "Yağız", "Bera"}, {"Matematik", "Tarih", "Yabancı
Dil", "Fizik", "Kimya"};

        for (int i=0; i<5; i++) {

            for (int j=0; j<5; j++) {
                System.out.print(ogrenciler[i][j]);
            }
            System.out.println ("");
        }
    }
}
```

JAVA'DA METOTLARIN KULLANIMI

public class metotlar1 {

```
public static void main(String[] args) {  
    //parametresiz ve geriye değer döndürmeyen metotlar  
    etiket();  
}
```

// Yeni bir metot için main metodunun dışındayız..

// Kod grubunu 1 kez yazıp istediğimiz yerde metot adı ile çağırıyoruz.

// erişim belirleyicisi (public) - dönüş tipi (staic void) - adı () (etiket())

```
public static void etiket() {  
    System.out.println("Ali KESKİN");  
    System.out.println("Sınıf: 10 / E ");  
    System.out.println("Numara: 107");    }}
```

public class metotlar2 {

```
public static void main(String[] args) {  
    //parametrelili () ve geriye değer döndürmeyen (void)metotlar  
    topla(34,12);  
    kullanıcı ("Ali",2010);  
}
```

// Programda "topla" adlı metodu çağırdığımız yerde hangi değişken bilgilerini almamız gerekiyorsa tanımlarız...

```
public static void topla(int sayi1, int sayi2) {  
    int sonuc= sayi1 + sayi2;  
    System.out.println("Sonuç:" +sonuc);  
}
```

//kullanıcı adlı bir metot tanımlayalım

```
public static void kullanıcı(String ad, int yıl) {  
    int yas = 2025 - yıl;  
    System.out.println(ad+"\t"+ yas +"\t"+ "yaşındasın...");  
}}
```

```
public class metotlar3 {
```

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

```
        //parametrelili () ve geriye değer döndüren metotlar
```

```
        // return anahtar kelimesini kullanıyoruz.
```

```
        // Dönen sonucu program içinde değişkene atayıp kullandık.
```

```
        System.out.println("Ders Etkinlikleri Ortalama Not:");
```

```
        double sonuc = ortalama(91,50);
```

```
        System.out.println(sonuc);
```

```
    }
```

```
// Geriye değer döndürmesi için void yerine metot sonunda oluşan değerini veri türü (double) yazılır.
```

```
    public static double ortalama (int sayi1, int sayi2) {
```

```
        double sonuc= (sayi1 + sayi2)/2;
```

```
        return(sonuc);
```

```
    }}
```

JAVA'DA BİRDEN FAZLA CLASS İLE ÇALIŞMA

com.alikeskin Klasörü – ogrenci.java Sınıfı

```
package com.alikeskin;

public class ogrenci {

    public int no;
    public String ad;
    public String soyad;
}
```

test Klasörü – test .java Sınıfı

```
package Test;

// ogrenci class ını kullanabilmek için import ederiz.
import com.alikeskin.ogrenci;

// ogrenci classını kullanmak için ogrenci yazdığımızda ALT ENTER ya da üzerine gelince IMPORT ile
// bu satırı ekleriz.

public class test {

    public static void main(String[] args) {

        //ogrenci classına erişebilmek için nesne türetmemiz gerek...
        ogrenci verigir = new ogrenci();

        // ogrenci classı - erişim için bir nesne ismi oluşturduk - new (anahtar kelime) - ogrenci() metodu
        //ogrenci sınıfından bir nesne türettik.

        verigir.no=107;
        verigir.ad="Veli";
        verigir.soyad="Keskin";

        //Bu bilgileri yazdıralım.

        System.out.println("ÖĞRENCİ BİLGİLERİ:");
        System.out.println("No:" + verigir.no);
        System.out.println("Ad:" + verigir.ad);
        System.out.println("Soyad:" + verigir.soyad);

    }

}
```

JAVA'DA GUI FORM OLUŐTURMA

Ad Soyad Bilgilerini alan bir Program Formu oluőturalım.