

Diziler

BİL131 - Bilişim Teknolojileri ve Programlama

Hakan Ezgi Kızılöz

Diziler

- Çok sayıda aynı türdeki değeri farklı isimlerdeki değişkenlerde saklamak yerine, dizi kullanarak tek bir değişken adı kullanarak saklayabiliriz.
- **Diziler, aynı veri türüne veya sınıfa ait birbirleriyle ilişkili veriler topluluğudur.**
- Dizilerin her bir elemanında saklanan veri basit veri türüne veya bir nesneye ait olabilir.
- Bellekte kaplayacağı boyut veri türüne veya nesneye göre belirlenir.

Diziler

- Aynı veri türüne ait n adet veri, indisleri 0'dan başlayan ve $(n-1)$ 'e kadar uzanan bir veriler dizisi içerisinde saklanabilir. Bahsedilen veriler dizisi aşağıdaki biçimde tanımlanmaktadır:

```
veri_turu[] veriler = new veri_turu[n];
```

- Örneğin 5 tane tam sayı saklayabilen `sayilar` dizisi aşağıdaki biçimde tanımlanmaktadır:

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

Diziler

- Örneğin Ankara'nın bir haftalık hava sıcaklık değerlerini dizi kullanmadan kullanan bir program yazmak istersek aşağıdaki gibi 7 farklı tam sayı değişkeni oluşturmamız gerekir:

```
int gun1, gun2, gun3, gun4, gun5, gun6, gun7;
```

- Oysa bütün bu değerleri sadece bir dizi içerisinde tutabiliriz:

```
int[] sicakliklar = new int[7];
```

Diziye Değer Atama

- Her iki örnek için de tanımlanmış değişkene değer atama işlemi benzer şekilde gerçekleşir:

<code>gun1 = 7;</code>	<code>sicaklik[0] = 7;</code>
<code>gun2 = 4;</code>	<code>sicaklik[1] = 4;</code>
<code>gun3 = -1;</code>	<code>sicaklik[2] = -1;</code>
<code>gun4 = -3;</code>	<code>sicaklik[3] = -3;</code>
<code>gun5 = 0;</code>	<code>sicaklik[4] = 0;</code>
<code>gun6 = 2;</code>	<code>sicaklik[5] = 2;</code>
<code>gun7 = 3;</code>	<code>sicaklik[6] = 3;</code>

- Eğer dizi elemanlarının alacakları değerler önceden belliyse değer ataması dizi oluşturulurken de yapılabilir.

```
int[] sicaklik = {7, 4, -1, -3, 0, 2, 3};
```

Diziler

- Ortalama hesabı yapmak istersek, dizi kullanmadığımız durumda bütün değişkenleri teker teker toplamalı ve sonrasında toplam değişken sayısına bölmeliyiz.

```
double ortalama = (gun1+gun2+gun3+gun4+gun5+gun6+gun7)/7.0;
```

- Oysa dizi kullanırsak ortalama hesabını basit bir döngüyle yapabiliriz:

```
int toplam = 0;  
for(int i = 0; i < 7; i++)  
    toplam += sicakliklar[i];
```

```
double ortalama = toplam / 7.0;
```

Dizinin Boyu

- Dizinin boyunu `dizinin_ismi.length` komutu ile otomatik olarak bulabiliriz.

```
double notlar = new notlar[117];  
System.out.print("Dersi "+ notlar.length + " ogrenci aliyor.");
```

- Bir dizinin boyutu, dizinin `length` değişkenine atama yaparak değiştirilemez.

```
notlar.length = 100; // yazilamaz
```

- Bir dizinin boyutundan sonraki indislere erişilemez.

```
notlar[117] = 36.5; // Maksimum indis asildi!!
```

Dizi Elemanlarının Varsayılan Değeri

- Bir dizi tanımlanmış fakat dizi elemanlarına değer atanmamışsa, her tür için bir adet varsayılan değer otomatik olarak atanmaktadır.
- Bu varsayılan değer tam sayılar için `0`, ondalıklı sayılar için `0.0`, `boolean` için `false`, `char` için boşluk karakteri ve `String` için `null` değeridir.

Dizileri Kopyalamak

- Yapısı gereği, bir dizi elemanlarının değerlerini başka bir dizinin elemanlarının değerleri olarak belirlemek için; yani bir diziyi başka bir diziye kopyalamak için klasik yöntemle atama işlemi yapmamalıyız.
- Örneğin aşağıdaki kod parçasını inceleyelim:

```
int[] a = {1, 2, 3};  
int[] b = a;  
b[0] = 4;  
System.out.println(a[0]);
```

Dizileri Kopyalamak

- Bir dizi içeriğini başka bir diziye kendisi daha sonra değişikliklerden etkilenmeyecek şekilde kopyalamak istiyorsak, bunu iki yolla yapabiliriz:
 - 1) Bir döngü içerisinde ilk dizinin bütün elemanlarını dolaşır ve her birini ikinci diziye değer olarak atarız.

```
int[] a = {1, 2, 3};  
int[] b = new int[a.length];  
for(int i = 0; i < a.length; i++)  
    b[i] = a[i];
```

- 2) Diziler için tanımlı olan `.clone()` metodunu kullanırız.

```
int[] a = {1, 2, 3};  
int[] b = a.clone();
```

Dizileri Metotlara Göndermek

- Bir dizinin bir, birkaç elemanını veya tamamını bir metot içerisine parametre olarak gönderebilirsiniz.
- Örneğin `int[] sayilar = {1, 2, 3, 4, 5};` komutu ile `sayilar` dizisi tanımlanmış olsun.
- `sayilar` dizisinin her bir elemanı aslında bir tam sayı olduğundan, parametre olarak bir tam sayı alan metotlara parametre olarak gönderilebilirler.
- Örneğin parametre olarak iki tane tam sayı alan ve bu tam sayıların toplamını hesaplayarak geri döndüren `ikiSayiTopla(int, int)` metodu ile parametre olarak bir tam sayı dizisi alan ve dizinin bütün elemanlarının toplamını hesaplayarak geri döndüren `diziTopla(int[])` metotlarımız yazılmış olsun.

Dizileri Metotlara Göndermek

- Eğer sayılar dizisinin sadece ilk ve son elemanlarının toplamını hesaplamak istiyorsak `ikiSayiTopla(sayilar[0], sayilar[4]);` komutu ile sadece bu sayıların toplamını hesaplayan metodu çağırabiliriz.
- Eğer dizi elemanlarının tamamının toplamını hesaplamak istiyorsak `diziTopla(sayilar);` komutunu kullanmamız gerekir.

Dizilerin Eşitliği

- String karşılaştırmasında olduğu gibi, iki dizinin birbirleri ile aynı olup olmadıklarını kontrol etmek için `==` karşılaştırmasını kullanamayız.
- Örneğin şu program parçasını inceleyelim:

```
int[] a = {1, 2, 3};  
int[] b = {1, 2, 3};  
if(a == b) System.out.println("diziler esit");  
else System.out.println("diziler esit degil");
```

Dizilerin Eşitliği

- Oysa biliyoruz ki her iki dizi de 3 elemandan oluşmakta ve her iki dizinin de elemanları sırasıyla 1, 2 ve 3 tam sayıları.
- Dizilerde eşitlik kontrolü yapabilmek için kendi `boolean esitMi(veri_turu[], veri_turu[])` metodumuzu yazmamız gerekmektedir. Bu metot parametre olarak aynı türde iki dizi almalıdır. Metot, eğer her iki dizinin de boyutları aynı ve dizilerin aynı sıradaki elemanlarının değerleri aynıysa geriye `true` değerini; bahsedilen koşullardan bir tanesi bile sağlanmazsa geriye `false` değerini döndürmelidir.

Dizilerin Eşitliği

- Biraz önce hatalı şekilde karşılaştırdığımız a ve b dizilerinin eşitlik kontrolünü yapabilmek için boolean `esitMi(int[], int[])` metodunu yazalım.

```
public static boolean esitMi(int[] ilk, int[] ikinci){  
    if(ilk.length != ikinci.length) return false;  
    for(int i = 0; i < ilk.length; i++){  
        if(ilk[i] != ikinci[i]) return false;  
    }  
    return true;  
}
```

Dizi Döndüren Metotlar

- Bir metot parametre olarak bir dizi alabileceği gibi, metot içerisinde bir dizi oluşturduktan sonra geriye diziyi de döndürebilmektedir.
- Örneğin parametre olarak pozitif bir n tam sayısı alan ve elemanları 0'dan $(n-1)$ 'e kadar değişen n elemanlı bir tam sayı dizisini oluşturarak geri döndüren `diziOlustur(int)` metodu aşağıdaki gibi tanımlanabilir:

```
public static int[] diziOlustur(int sayi){  
    int[] dizi = new int[sayi];  
    for(int i = 0; i < sayi; i++)  
        dizi[i] = i;  
    return dizi;  
}
```