



Java'da Dizi (“Array”) ve Dizgi (“String”) Tanımlama

BBS-515 Nesneye Yönelik Programlama

Ders #6 (25 Kasım 2009)

■ Geçen ders:

- ▶ Kalıtım (“inheritance”) -- tekrar ziyaret
- ▶ Java’da “super” kullanımı
- ▶ Java’da “static” ve “final” tanımları
- ▶ Java’da kapsülleme (“encapsulation”)

■ Bu ders:

- ▶ Java’da dizi (“array”) ve dizgi (“String”) tanımlama
- ▶ Örnekler

Dizi (“Array”) Tanımlama - 1

■ Dizi aslında bir nesnedir.

■ Sözdizimi:

▶ *type [] degisken_adi;* veya *type degisken_adi [];*

■ Örnek:

▶ `int [] aygun;` // ayın günlerini tutmak için tamsayı dizisi

▶ `aygun = new int [12];`

▶ İkisi birarada: `int [] aygun = new int [12];`

▶ `aygun [1] = 28;` // 0-tabanlı endeksleme

■ İklendirme:

▶ `int [] aygun = { 31, 28, 31, 30, ...};`

Dizi: Örnek – 1

```
class UzunlukDemo {  
    public static void main (String args []) {  
        int [] a1 = new int [10];  
        int [] a3 = {4,3,2,1};  
        // ilklendirme tanımlama aşamasında yapılır.  
        // a3 = {4,3,2,1}; ilklendirmesine izin verilmez.  
        System.out.println (“a3 uzunluğu : ” + a3.length);  
    }  
}
```

Dizi: Örnek – 2

```
int [] a = new int [6];
```

```
int [] b = {13, 23, 55, 4};
```

```
a = b; // sadece bellek adresini (referansı) kopyalar.
```

```
a = (int[]) b.clone(); // dizinin aynısını kopyalar.
```

tip değiştirme nesne ("object")
(type casting) döndürür

Nesneyi tamsayı dizisine çevirerek geçirir.

Dizi (“Array”) Tanımlama - 2

- `type [] f;` // Diziye referans tanımlar.
- `type [] f = new type [number];` // Diziyi tanımlar ve ilkendirir.
- `type [] f = {value, value, ..., value};` // Diziyi tanımlar ve ilkendirir.
- `f = new type [] {value, value, ..., value};`
// Yeni bir diziyi tanımlayıp ilkendirerek f'e atar.
- `f = g;` // f, g dizisine işaret (referans) eder.
- `f = (type[]) g.clone();` // f, g dizisinin kopyasına işaret (referans) eder.
- `f.length;` // f dizisindeki elemanların sayısını döndürür.
- `System.arraycopy (f1, startpos1, f2, startpos2, number);`
// *number* ile belirtilen sayıdaki elemanı, f1 dizisinin *startpos1*
// elemanından itibaren, f2 dizisine *startpos2* elemanından başlayarak kopyalar.
- `f == g`
// f ve g dizi değişkenlerinin, aynı diziye referans edip etmediklerini kontrol eder.

```
public class DiziDemo {
    public static void main (String args []) {
        String[] h1;           // Diziyi referans tanımlar.
        String[] h2 = new String[7]; // Diziyi tanımlar ve ilklendirir.
        // Diziyi tanımlar ve ilklendirir.
        String[] h3 = {"Pazartesi", "Salı", "Çarşamba", "Perşembe", "Cuma", "Cumartesi", "Pazar"};
        // Yeni bir diziyi tanımlayıp ilklendirerek h4'e atar.
        String[] h4 = new String [] {"Pazartesi", "Salı", "Çarşamba", "Perşembe", "Cuma", "Cumartesi", "Pazar"};

        // h1, h3 dizisine işaret (referans) eder.
        h1 = h3;
        h1[0]= "P.tesi"; h1[5] = "C.tesi";

        // h1 ve h3 dizi değişkenlerinin, aynı diziyi referans edip etmediklerini kontrol eder.
        if (h1 == h3) System.out.println ("h1 ve h3 aynı diziyi işaret ediyor.");
            else System.out.println ("h1 ve h3 aynı diziyi işaret etmiyor.");

        // h1, h4 dizisinin kopyasına işaret (referans) eder.
        h1 = (String[]) h4.clone();
        h1[0]= "P.tesi"; h1[5] = "C.tesi";

        // h1 ve h3 dizi değişkenlerinin, aynı diziyi referans edip etmediklerini kontrol eder.
        if (h1 == h3) System.out.println ("h1 ve h3 aynı diziyi işaret ediyor.");
            else System.out.println ("h1 ve h3 aynı diziyi işaret etmiyor.");

        // f dizisindeki elemanların sayısını döndürür.
        System.out.println("h1 dizisinin uzunluğu: " + h1.length);

        // h4 dizisinin son elemanını h2 dizisinin ilk elemanı olarak atar.
        h2[0]=h4[6];

        // System.arraycopy (f1, startpos1, f2, startpos2, number);
        // number ile belirtilen sayıdaki elemanı, f1 dizisinin startpos1
        // elemanından itibaren, f2 dizisine startpos2 elemanından başlayarak kopyalar.
        // h2 dizisinin kalan elemanlarına, h4 dizisinin ilk 6 elemanını kopyalar.
        System.arraycopy(h4, 0, h2, 1, 6);

        // h2 dizisinin elemanlarını yazdırır.
        System.out.println("h2 dizisinin elemanları: ");
        for (int i=0; i<7; i++)
            System.out.println (h2[i]);
    }
}
```

Çok-Boyutlu Dizi Tanımlama

■ Tanım:

▶ `int ikiBoyutlu [] [] = new int [4] [5];`

■ Java'da çok boyutlu bir dizinin satır ve sütun sayıları aynı olmak zorunda değildir.

■ Örnek:

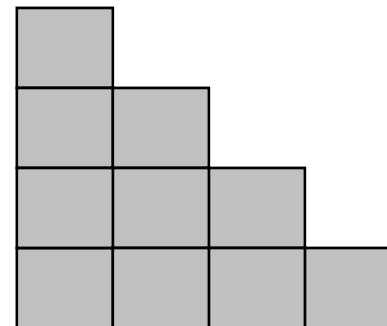
▶ `int ikiB [] [] = new int [4] [];`

▶ `ikiB [0] = new int [1];`

▶ `ikiB [1] = new int [2];`

▶ `ikiB [2] = new int [3];`

▶ `ikiB [3] = new int [4];`



Dizi: Alıştırma

- İlk 100 asal sayıyı bulan ve ekrana bastıran Java konsol programını yazın.

Dizgi (“String”) Tanımlama

- Dizgi aslında bir nesnedir.
- Tanım:
 - ▶ *String myString = “Bu örnek bir dizgidir.”;*
- “+” işleci dizgileri birleştirmek için kullanılır.
 - ▶ *String myString = “Bu örnek bir ” + “dizgidir.”;*
- “java.lang” paketi (“package”) içinde yer alır.
 - ▶ *import java.lang.String;*

Dizgi: Örnek – 1

```
class DizgiOrnek {  
    public static void main (String args[]) {  
        for (int i=0; i<args.length(); i++)  
            System.out.println (“args[“+ i + “]: ” + args[i]);  
        }  
    }  
}
```

- *javac DizgiOrnek.java*
- *java DizgiOrnek bu bir dizgi ornegidir 100 -1*

Çıktı:

```
args[0]: bu  
args[1]: bir  
args[2]: dizgi  
args[3]: ornegidir  
args[4]: 100  
args[5]: -1
```

“java.lang.String” Sınıfının Yöntemleri

```
String s1 = “Java ”, s2 = “Programming”;
```

```
char c = “g”, d = “s”;
```

- `s1.length();` // s1 içindeki karakter sayısını döndürür.
- `s1.charAt(n);` // s1 içindeki n. karakteri döndürür.
- `s1.equals(s2);` // s1 s2'ye eşitse 1, değilse 0 döndürür.
- `s2.indexOf(c);` // s1 içinde c ile tanımlı karakterin bulunduğu ilk sırayı
// döndürür; karakter yoksa -1 döndürür.
- `s2.lastIndexOf(c);` // s1 içinde c ile tanımlı karakterin bulunduğu son sırayı
// döndürür; karakter yoksa -1 döndürür.
- `s1.trim();` // s1 dizisinin başındaki ve sonundaki boşluk ve “tab”
// karakterlerini kaldırarak kopyasını döndürür.
- `s2.replace(c,d);` // s1 dizisi içindeki c ile tanımlı her karakteri,
// d ile tanımlı karakterle değiştirerek diziyi döndürür.

Dizgi: Örnek – 2

```
public class DizgiDemo {
    public static void main (String args []) {
        String s1 = "Java ";
        String s2 = "Programming";
        String s3;
        char c = 'g';
        char d = 's';

        // s1 içindeki karakter sayısını döndürür.
        System.out.println (s1 + " dizgisinin uzunluğu: " + s1.length());

        // s1 içindeki n. karakteri döndürür.
        System.out.println (s1 + " dizgisinin n. karakteri: " + s1.charAt(2));

        s3 = "Java ";
        // s1 s3'e eşitse 1, değilse 0 döndürür.
        if (s1.equals(s3))
            System.out.println ("s1 s3'e esit: " + s1 + " = " + s3);

        // s1 içinde c ile tanımlı karakterin bulunduğu ilk sırayı
        // döndürür; karakter yoksa -1 döndürür.
        System.out.println (s2 + " dizgisi icinde '" + c + "' karakterinin ilk konumu: " + s2.indexOf(c));
        System.out.println (s2 + " dizgisi icinde '" + d + "' karakterinin ilk konumu: " + s2.indexOf(d));

        // s1 içinde c ile tanımlı karakterin bulunduğu son sırayı
        // döndürür; karakter yoksa -1 döndürür.
        System.out.println (s2 + " dizgisi icinde '" + c + "' karakterinin son konumu: " + s2.lastIndexOf(c));
        System.out.println (s2 + " dizgisi icinde '" + d + "' karakterinin son konumu: " + s2.lastIndexOf(d));

        System.out.println(s1 + "6");
        // s1 dizisinin başındaki ve sonundaki boşluk ve "tab"
        // karakterlerini kaldırarak kopyasını döndürür.
        System.out.println(s1.trim() + "6");

        System.out.println(s2);
        // s1 dizisi içindeki c ile tanımlı her karakteri,
        // d ile tanımlı karakterle değiştirerek diziyi döndürür.
        System.out.println(s2.replace(c,d));
    }
}
```

Dizgi: Alıřtırma

- Verilen karakter dizisinin (dizginin) palindrome olup olmadıđını bulan Java konsol programını yazın.