

Java Dilinde Array Kavramı

İki şekilde array tanımı vardır:

- 1) datatype[] **arrayRefVar**;
arrayRefVar=new datatype[];
- 2) datatype[] **arrayRefVar=new** datatype[];

Hatırlarsak **new** ile yaratılan her şey referanstır. C dilindeki karşılığı ise pointer'dır. Java'da array'ler referans değişkenler vasıtasıyla kullanılır. C dilinde array'ler ise pointer özelliğine sahiptir.

1. şekildeki gibi tanımlama C dilinde yoktur. Bu tür deklare edilme nedeni, programın ortalarında veya sonlarında array değişkeninin eleman sayısını belirlemeye olanak sağlamasından dolayıdır.

Örnek:

```
1 import java.util.Scanner;
2 public class ArrayDeclaration{
3     public static void main(String[] args){
4         int[] values;
5         System.out.println("Lütfen bir değer giriniz: ");
6         Scanner input=new Scanner(System.in);
7         int y=input.nextInt();
8         if(y==10){
9             values=new int[y];
10        }else{
11            values=new int[0];
12        }
13        System.out.println("Oluşturulan array'in boyutu: "+values.length);//Ekrana 10 yazar
14    }
15 }
```

Örnekte görüldüğü gibi, if içerisindeki şart doğru olduğunda array'in eleman sayısı 10 oluyor.

2. şekildeki gibi tanımlama C dilindeki array tanımlaması ile aynıdır. Bu deklarasyonu da programın herhangi bir yerinde yapabiliriz. C dilinde ise sadece fonksiyonun başında yazabiliriz.

For-each Döngüsü

```
1 double myList={1.9,2.9,3.4,3.5};
2 for(double element:myList){
3     System.out.println(element);
4 }
```

Element demek myList[0], myList[1]... demektir. myList array'inin tüm elemanları ekrana yazdırılır.

For-each'in faydası şudur: Eleman sayısını bilmediğimiz bir array'in elemanlarını yazdırmaya olanak tanır.

Array'leri Kopyalamak

```
1 double[] a=new double[10];
2 double[] b=new double[10];
```

a=b demekle b'nin değerleri a'ya atanmaz. b değişkeninin **tuttuğu adres** a'ya **atanır**. Çünkü a,b referans değişkenlerdir. Eğer b'nin elemanlarını a'nın elemanlarına atamak istiyorsak **üç yol** ile bunu yapabiliriz:

1. **for** döngüsüyle
2. **System** sınıfının **arrayCopy()** isimli **static** metoduyla
3. **clone()** metodu ile.

```
1 System.arrayCopy(b,0,a,0,3);
```

arrayCopy() metodunun ilk parametresi kaynak array, ikinci parametre kaynak array'in başlangıç pozisyonu, üçüncü parametre hedef array, dördüncü parametre hedef array'inin başlangıç pozisyonu, beşinci parametre kaç tane eleman kopyalanacağını ifade eder.

clone() metodu **Object** sınıfının bir metodudur. Bu metod ile kopyalama işlemi şu şekilde olur:

```
1 int[] dest = b.clone();
```

clone metodu ile kopyalama işlemi yapıldığı zaman dest değişkeni b array'i ile aynı boyuta aynı değerlere sahip olur. Bu metod esnek olmadığı için ve daha yavaş çalıştığı için, array kopyalama işlemi için tavsiyem **System.arrayCopy()** metodu kullanılması yönündedir.

Array'i Metod Parametresi Olarak Kullanmak

Bir metoda array "pass" etmek çok önemlidir.

```
1 public static void printArray(int[] array){
2     for(int i=0; i<array.length; i++){
3         System.out.println(array[i]+" ");
4     }
}
```

```
5 | }
```

Bir array'in uzunluğu arrayAdi.length ile bulunur. Görüldüğü gibi parametre içerisindeki parametre türü int[], parametre ismi array olan bir static metod tanımladık.

```
1 | public static void main(String[] args){
2 |     int[] dizi={1,2,3};
3 |     printArray(dizi);
4 | }
```

Bir Metoddan Array Döndürmek

```
1 | public static int[] reverse(int[] list){
2 |     int[] result=new int[list.length];
3 |     for(int i=0; j=result.length-1; i<list.length; i++,j--){
4 |         result[j]=list[i];
5 |     }
6 |     return result;
7 | }
```

reverse metodu bir array alır ve bir array döndürür.

İki Boyutlu Array

```
datatype[][] arrayRefVar=new datatype[][];
```

veya

```
datatype[][] arrayRefVar;
arrayRefVar=new datatype[][];
```

şeklinde tanımlanır.

Örnek:

```
1 | double[][] list=new double[5][3];
2 | int a=list[0].length;//a ya 3 değeri atanır.
3 | double[][] list=new int[2][];
4 | list[0]=new int[5]; //ilk satır uzunluğu 5
5 | list[1]=new int[4]; //ikinci satır uzunluğu 4
```