**Diziler ARRAYS**

*Dizi* , tek bir türden sabit sayıda değeri tutan bir kapsayıcı nesnedir. Bir dizinin uzunluğu, dizi oluşturulduğunda belirlenir. Oluşturulduktan sonra uzunluğu sabittir. main"Merhaba Dünya!" yönteminde zaten dizilerin bir örneğini gördünüz. başvuru. Bu bölümde diziler daha ayrıntılı olarak ele alınmaktadır.



10 elemanlı bir dizi.

Bir dizideki her öğeye öğe adı verilir *ve* her öğeye sayısal *dizini* ile erişilir . Önceki çizimde gösterildiği gibi, numaralandırma 0 ile başlar. Bu nedenle, örneğin 9. öğeye 8. indeksten erişilebilir.

Aşağıdaki program, [ArrayDemo](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/examples/ArrayDemo.java" \t "_blank)bir tamsayı dizisi oluşturur, diziye bazı değerler koyar ve her değeri standart çıktıya yazdırır.

sınıf ArrayDemo {

public static void main(String[] args) {

// bir tamsayı dizisi bildirir

int[] anArray;

// 10 tam sayı için bellek ayırır

anArray = yeni int[10];

// ilk elemanı başlat

anArray[0] = 100;

// ikinci elemanı başlat

anArray[1] = 200;

// ve benzeri

anArray[2] = 300;

anArray[3] = 400;

anArray[4] = 500;

anArray[5] = 600;

anArray[6] = 700;

anArray[7] = 800;

anArray[8] = 900;

anArray[9] = 1000;

System.out.println("İndeks 0'daki eleman: "

+ anArray[0]);

System.out.println("İndeks 1'deki öğe: "

+ anArray[1]);

System.out.println("İndeks 2'deki eleman: "

+ anArray[2]);

System.out.println("İndeks 3'teki öğe: "

+ birArray[3]);

System.out.println("İndeks 4'teki eleman: "

+ birArray[4]);

System.out.println("İndeks 5'teki eleman: "

+ birArray[5]);

System.out.println("İndeks 6'daki eleman: "

+ birArray[6]);

System.out.println("İndeks 7'deki eleman: "

+ birArray[7]);

System.out.println("İndeks 8'deki eleman: "

+ birArray[8]);

System.out.println("İndeks 9'daki eleman: "

+ birArray[9]);

}

}

Bu programın çıktısı:

0: 100 dizinindeki öğe

İndeks 1: 200'deki eleman

Dizin 2: 300'deki öğe

Dizin 3: 400'deki öğe

Dizin 4: 500'deki öğe

Dizin 5'teki eleman: 600

Dizin 6'daki eleman: 700

Dizin 7'deki öğe: 800

İndeks 8'deki eleman: 900

Dizin 9'daki eleman: 1000

Gerçek dünya programlama durumunda, önceki örnekte olduğu gibi her satırı ayrı ayrı yazmak yerine, dizinin her bir öğesini yinelemek için muhtemelen desteklenen *döngü yapılarından birini kullanırsınız.*Bununla birlikte, örnek, dizi sözdizimini açıkça göstermektedir. [Kontrol Akışı](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/flow.html) bölümünde çeşitli döngü yapıları ( for, while, ve do-while) hakkında bilgi edineceksiniz .

**Bir Diziye Başvuracak Bir Değişken Bildirmek**

anArrayÖnceki program , aşağıdaki kod satırıyla bir dizi (adlı) bildirir :

// bir tamsayı dizisi bildirir

int[] anArray;

Diğer türlerdeki değişkenler için bildirimler gibi, bir dizi bildiriminin iki bileşeni vardır: dizinin türü ve dizinin adı. Bir dizinin türü olarak yazılır , içerdiği öğelerin veri türü *type*[]burada ; *type*parantezler, bu değişkenin bir diziyi tuttuğunu gösteren özel sembollerdir. Dizinin boyutu, türünün bir parçası değil (bu nedenle parantezler boştur). Bir dizinin adı, daha önce [adlandırma](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html#naming) bölümünde tartışıldığı gibi kurallara ve kurallara uyması koşuluyla, istediğiniz herhangi bir şey olabilir. Diğer türlerdeki değişkenlerde olduğu gibi, bildirim aslında bir dizi oluşturmaz; derleyiciye bu değişkenin belirtilen türde bir dizi tutacağını söyler.

Benzer şekilde, diğer türlerin dizilerini de bildirebilirsiniz:

byte[] anArrayOfBytes;

kısa[] anArrayOfShorts;

uzun[] anArrayOfLongs;

float[] anArrayOfFloats;

double[] anArrayOfDoubles;

boolean[] anArrayOfBooleans;

karakter[] anArrayOfChars;

String[] anArrayOfStrings;

Parantezleri dizinin adından sonra da yerleştirebilirsiniz:

// bu form önerilmez

float anArrayOfFloats[];

Ancak, sözleşme bu biçimi önermemektedir; parantezler dizi tipini tanımlar ve tip tanımıyla birlikte görünmelidir.

**Dizi Oluşturma, Başlatma ve Diziye Erişme**

Dizi oluşturmanın bir yolu newoperatördür. Programdaki bir sonraki ifade, 10 tamsayı elemanı için yeterli belleğe sahip bir dizi ayırır ve diziyi değişkene ArrayDemoatar .anArray

// bir tamsayı dizisi oluştur

anArray = yeni int[10];

Bu ifade eksikse, derleyici aşağıdaki gibi bir hata yazdırır ve derleme başarısız olur:

ArrayDemo.java:4: anArray değişkeni başlatılmamış olabilir.

Sonraki birkaç satır, dizinin her bir öğesine değerler atar:

anArray[0] = 100; // ilk elemanı başlat

anArray[1] = 200; // ikinci elemanı başlat

anArray[2] = 300; // ve benzeri

Her dizi elemanına kendi sayısal indeksi ile erişilir:

System.out.println("İndeks 0'daki Öğe 1: " + anArray[0]);

System.out.println("İndeks 1'deki Öğe 2: " + anArray[1]);

System.out.println("İndeks 2'deki Öğe 3: " + anArray[2]);

Alternatif olarak, bir dizi oluşturmak ve başlatmak için kısayol sözdizimini kullanabilirsiniz:

int[] anArray = {

100, 200, 300,

400, 500, 600,

700, 800, 900, 1000

};

Burada dizinin uzunluğu, parantezler arasında sağlanan ve virgülle ayrılan değerlerin sayısı ile belirlenir.

gibi iki veya daha fazla parantez kümesi kullanarak bir dizi dizisini ( *çok boyutlu* dizi olarak da bilinir) bildirebilirsiniz String[][] names. Bu nedenle, her öğeye karşılık gelen sayıda dizin değeriyle erişilmelidir.

Java programlama dilinde, çok boyutlu bir dizi, bileşenleri dizi olan bir dizidir. Bu, C veya Fortran'daki dizilerden farklıdır. [MultiDimArrayDemo](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/examples/MultiDimArrayDemo.java" \t "_blank)Bunun bir sonucu, aşağıdaki programda gösterildiği gibi, satırların uzunluklarının değişmesine izin verilmesidir :

sınıf MultiDimArrayDemo {

public static void main(String[] args) {

Dize[][] adları = {

{"Bay Bayan Bayan. "},

{"Smith", "Jones"}

};

// Bay Smith

System.out.println(isimler[0][0] + isimler[1][0]);

// Bayan Jones

System.out.println(isimler[0][2] + isimler[1][1]);

}

}

Bu programın çıktısı:

Bay Smith

Bayan Jones

lengthSon olarak, herhangi bir dizinin boyutunu belirlemek için yerleşik özelliği kullanabilirsiniz. Aşağıdaki kod, dizinin boyutunu standart çıktıya yazdırır:

System.out.println(anArray.length);

**Dizileri Kopyalama**

Sınıf , verileri bir diziden diğerine verimli bir şekilde kopyalamak için kullanabileceğiniz Systembir yönteme sahiptir :arraycopy

genel statik geçersiz dizi kopyası(Nesne kaynağı, int kaynakPos,

Nesne hedef, int hedefPos, int uzunluk)

İki bağımsız değişken , kopyalanacak diziyi ve *kopyalanacak*Object diziyi belirtir *.*Üç bağımsız değişken, kaynak dizideki başlangıç ​​konumunu, hedef dizideki başlangıç ​​konumunu ve kopyalanacak dizi öğelerinin sayısını belirtir.int

Aşağıdaki program, [ArrayCopyDemo](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/examples/ArrayCopyDemo.java" \t "_blank), bir dizi Stringeleman bildirir. System.arraycopyDizi bileşenlerinin bir alt dizisini ikinci bir diziye kopyalamak için yöntemi kullanır :

class ArrayCopyDemo {

public static void main(String[] args) {

String[] copyFrom = {

"Affogato", "Americano", "Cappuccino", "Corretto", "Cortado",

"Doppio", "Espresso", "Frappucino", "Freddo", "Lungo", "Macchiato",

"Marocchino", "Ristretto" };

String[] copyTo = new String[7];

System.arraycopy(copyFrom, 2, copyTo, 0, 7);

for (String kahve: copyTo) {

System.out.print(kahve + " ");

}

}

}

Bu programın çıktısı:

Cappuccino Corretto Cortado Doppio Espresso Frappucino Freddo

**Dizi Manipülasyonları**

Diziler, programlamada kullanılan güçlü ve kullanışlı bir kavramdır. Java SE, dizilerle ilgili en yaygın manipülasyonlardan bazılarını gerçekleştirmek için yöntemler sağlar. Örneğin, örnek , kaynak dizinin öğelerini manuel olarak yinelemek ve her birini hedef diziye yerleştirmek yerine sınıfın yöntemini [ArrayCopyDemo](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/examples/ArrayCopyDemo.java" \t "_blank)kullanır . Bu, arka planda gerçekleştirilir ve geliştiricinin yöntemi çağırmak için yalnızca bir kod satırı kullanmasını sağlar.arraycopySystem

Size kolaylık sağlamak için Java SE, sınıfta dizi manipülasyonları (dizileri kopyalama, sıralama ve arama gibi genel görevler) gerçekleştirmek için çeşitli yöntemler sağlar [java.util.Arrays](https://docs.oracle.com/javase/8/docs/api/java/util/Arrays.html" \t "_blank). Örneğin, örnekte görebileceğiniz gibi, önceki örnek , sınıfın copyOfRangeyöntemini kullanacak şekilde değiştirilebilir . Aradaki fark, hedef dizi yöntem tarafından döndürüldüğünden, yöntemi kullanmanın yöntemi çağırmadan önce hedef diziyi oluşturmanızı gerektirmemesidir:java.util.Arrays[ArrayCopyOfDemo](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/examples/ArrayCopyOfDemo.java)copyOfRange

class ArrayCopyOfDemo {

public static void main(String[] args) {

String[] copyFrom = {

"Affogato", "Americano", "Cappuccino", "Corretto", "Cortado",

"Doppio", "Espresso", "Frappucino", "Freddo", "Lungo", "Macchiato",

"Marocchino", "Ristretto" };

String[] copyTo = Java.util.Arrays.copyOfRange(copyFrom, 2, 9);

for (String kahve: copyTo) {

System.out.print(kahve + " ");

}

}

}

Gördüğünüz gibi, daha az kod satırı gerektirmesine rağmen, bu programın çıktısı aynıdır. Yöntemin ikinci parametresinin copyOfRange, dahil olarak kopyalanacak aralığın ilk dizini olduğunu, üçüncü parametrenin ise *yalnızca* kopyalanacak aralığın son dizini olduğunu unutmayın . Bu örnekte, kopyalanacak aralık, dizin 9'daki (dizeyi içeren) dizi öğesini içermez Lungo.

java.util.ArraysSınıftaki yöntemler tarafından sağlanan diğer bazı yararlı işlemler şunlardır:

* Bir diziyi, yerleştirildiği dizini ( binarySearchyöntem) almak için belirli bir değer için arama.
* Eşit olup olmadıklarını belirlemek için iki diziyi karşılaştırma ( equalsyöntem).
* fillHer dizine ( yöntem) belirli bir değer yerleştirmek için bir diziyi doldurma .
* Bir diziyi artan düzende sıralama. Bu, yöntem kullanılarak sıralı olarak sortveya Java SE 8'de tanıtılan yöntem kullanılarak eşzamanlı olarak yapılabilir parallelSort. Çok işlemcili sistemlerde büyük dizilerin paralel olarak sıralanması, sıralı dizi sıralamadan daha hızlıdır.
* stream Kaynağı ( yöntem) olarak bir dizi kullanan bir akış oluşturma . Örneğin, aşağıdaki ifade copyTodizinin içeriğini önceki örnekte olduğu gibi yazdırır:
* java.util.Arrays.stream(copyTo).map(kahve -> kahve + " ").forEach(System.out::print);

Akışlar hakkında daha fazla bilgi için [Toplama İşlemleri'ne](https://docs.oracle.com/javase/tutorial/collections/streams/index.html) bakın .

* Bir diziyi bir dizgeye dönüştürmek. Yöntem toString, dizinin her bir öğesini bir dizgeye dönüştürür, onları virgülle ayırır ve ardından parantez içine alır. Örneğin, aşağıdaki ifade copyTodiziyi bir dizgeye dönüştürür ve yazdırır:
* System.out.println(java.util.Arrays.toString(copyTo));

Bu ifade aşağıdakileri yazdırır:

[Cappuccino, Corretto, Cortado, Doppio, Espresso, Frappucino, Freddo]