**İçindekiler :**

1. **Array nedir?**
2. **Array’ın özellikleri?**
3. **Array olusturma nasıl yapılır?**
4. **Count yöntemi?**

**Arrays: birden fazla variable depolamak icin kullanılabilen object’lerdir**

1. **Arrays sadece**
   1. **Primitive data type**
   2. **Object Reference**

**içerebilir.**

* birden fazla varriable’i aynı anda kullanmak istediğimiz zaman arrays kullanıyoruz.
* Array’ler object’tir.
* Primitiveleri Java oluşturur. non primitive lardan ise istediğimiz kadar üretebiliriz. bu non primitivelerden biriside arraysdir.Arraysler heapte depolanır.
* heap büyük bir alan olduğu için bulunabilmesi için adreslenmesi gerekir.

1. **Arrays içindeki tum variable’lar aynı data type’ında olmalıdır.**

* arrays sadece Primitive type (byte, short...8 tane) ve objeckt referans konulabilir.
* arrays’ın içinde Stringin adresi olur, kendisi olmaz.
* array’in içine stackdekiler konulabilir.

1. **Bir Array oluşturmadan once o Array’in içine kaç variable koyacağımıza karar vermeliyiz. Bir Array icine koyabileceğimiz variable sayısına o Array’in “length” i denir. O Array icine length’den fazla variable koyamayız.**

* Array'in length'ini bulmak icin "ArrayIsmi.length" yazmak yeterlidir
* Stringlerde de length methodu kullanilir ama String'lerde kullanilan length methodu parantezlidir.
* Yani; String'lerde ==> .length() Array'lerde ==> .length

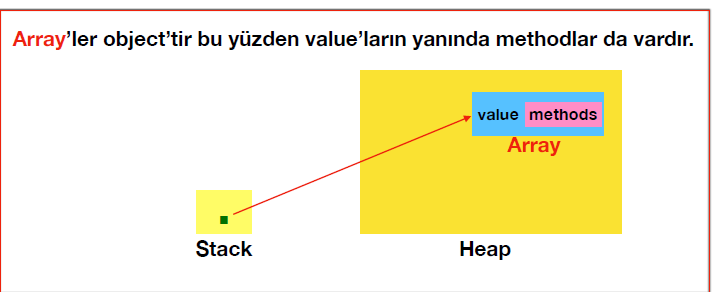
System.***out***.println(arr1.length);//4

ÖRNEK: Array'deki son elemani yazdirmak icin length kullaniniz:

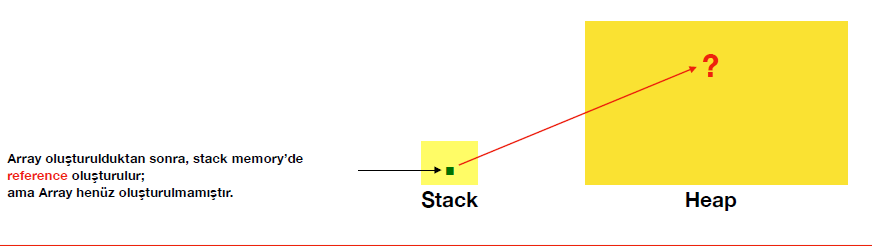
System.***out***.println(arr1[arr1.length-1]);//M

1. **Array’ler object’tir bu yüzden Heap Memory’de depolanırlar.**

* arrays sadece Primitive type (byte, short...8 tane) ve objeckt referans konulabilir.
* arrays’ın içinde Stringin adresi olur, kendisi olmaz.
* array’in içine stackdekiler konulabilir.

1. ****
2. **Array’ler object’tir bu yüzden runtime’a oluşturulurlar.**
3. **Array oluşturmak icin iki yol vardır : 1) int myArray[ ]; //Bu daha çok kullanılır**

**2) int [ ]myArray;**

1. ****
2. **int myArray[ ] = new int[6];**

**1) Yukarıdaki kod length’i 6 olan bir array oluşturur.**

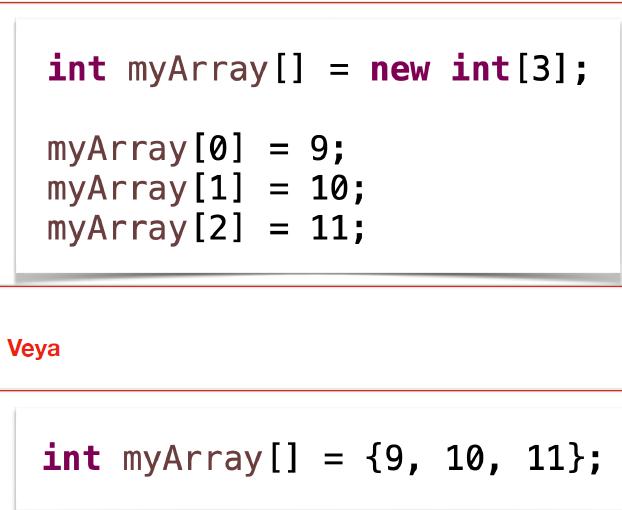
**2) biz array’e eleman eklemezsek Java elemanlar icin data type’ına uygun default degerler atar. Eğer yukarıdaki array’i yazdırırsanız ekranda {0, 0, 0, 0, 0, 0} gorursunuz**

1. **Array oluştururken length’i yazmazsanız compile time error alırsınız.**
2. **Integer Array olusturalim:**

**int arr1[] = new int[5];**

1. **System.out.println(arr1);==> Array'in reference'ini ekrana yazdirir**
2. **System.out.println(arr1[2]);==> Array'in index'i 2 olan elemanini ekrana yazdirir**

**------------------------ Bir Array’e Eleman Ekleme-------------------**

****

**ÖRNEK: Bir char array olusturalım:**

**char** arr1[] = **new** **char**[4];

arr1[0] = 'A';

arr1[1] = 'D';

arr1[2] = 'E';

arr1[3] = 'M';

**for**(**int** i=0; i<4; i++) {

System.***out***.print(arr1[i]); // ADEM

}

System.***out***.println(arr1[3]); //M

Soru: Elemanlari 1, 2, 3, 4, 5 olan bir integer array olusturunuz

Array olusturup Eleman eklemek icin 1. yol

int arr[] = new int[5];

arr[0]=1;

arr[1]=2;

arr[2]=3;

arr[3]=4;

arr[4]=5;

Daha kisa olan 2. yol:

**int** arr[]= {1,2,3,4,5};

**--------------COUNTER --------------**

Yukarıdaki array’de 3 elemaninin olup olmadigini kontrol ediniz.

3 varsa ekrana “3 var yazdirin”, 3 yoksa ekrana “3 yok yazdirin”

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

if (arr[i]==3) {

System.out.println("3 var");

}else {

System.out.println("3 yok");

}

}

Böyle yapabiliriz ama bu cok cirkin oldu onun yerine şunu yapalım:

**int** count = 0;

**for** (**int** i = 0; i <5; i++) {--🡪her bir elemanı kontrol için for döngüsüne ihtiyac var

**if** (arr[i]==3) {-------🡪eger arr[i] 3 ise count u bir artır.

count++; --------🡪her 3 degeri için count bir artırılır.

}

}

**if** (count>0) {

System.***out***.println("3 var");//her bir 3 degeri için count 1 arttı. Count sayısı kadar 3 var.

}**else** {

System.***out***.println("3 yok");

}

ÖNEMLİ:

* Array'de **olmayan index'e deger atamasi** yapilirsa **Run Time Error** alinir.
* Array'de olmayan indexi kullanmaya calisirsak **"ArrayIndexOutOfBoundsException"** aliriz.

--------------ARRAY 2.GÜN------------

1. soru:

Soru:

// Kullanıcıya kaç elemanlı bir array gireceğini sorun.

// Kullanıcıdan array’in elemanlarını girmesini isteyin.

// a) Bu array’in tum elemanlarını ekrana yazdırın.

// b) Bu arayın son elemanını ilk eleman yapın ve tum elemanlarını ekrana yazdırın.

// Mesela; array {1, 2, 3} ise ekrana {3, 1, 2} seklinde yazdırın

//a)

Scanner sc = **new** Scanner (System.***in***);

System.***out***.println("kac elemanli bir array olusturmak istersiniz?");

**int** elemanSayisi = sc.nextInt();

System.***out***.println("array in elamanlarini giriniz.");

**int** arr[] = **new** **int**[elemanSayisi];

**for** (**int** i = 0; i < elemanSayisi; i++) {

arr[i]=sc.nextInt();

}sc.close();

System.***out***.println(Arrays.*toString*(arr));

//b)

**int** arrNew[] = **new** **int**[elemanSayisi];

**for** (**int** i = 0; i < elemanSayisi-1; i++) {

arrNew[i-1] =arr[i];

}

arrNew[elemanSayisi-1]=arr[0];

System.***out***.println(Arrays.*toString*(arrNew));

2.Soru:

// Kullanıcıya kaç elemanlı bir array gireceğini sorun.

// Kullanıcıdan array’in elemanlarını girmesini isteyin.

// a) Bu array’in icinde herhangi bir elemanın var olup olmadigini kontrol edin.

// b) Bu arayın tum elemanlarını tersten yazdırın.

// Mesela; array {1, 2, 3, 4} ise ekrana {4, 3, 2, 1} seklinde yazdırın.

// a)

Scanner sc = **new** Scanner (System.***in***);

System.***out***.println("kac elemanlı bir array girmek istersiniz?");

**int** elemanSayisi = sc.nextInt();

System.***out***.println("array in elemanlarini giriniz.");

**int** arr[] = **new** **int**[elemanSayisi];

**for** (**int** i = 0; i < elemanSayisi; i++) {

arr[i]=sc.nextInt();

}sc.close();

System.***out***.println(Arrays.*toString*(arr));

// b)

**int** arrNew[] = **new** **int**[elemanSayisi];

**for** (**int** i=elemanSayisi-1; i >=0; i--) {

arrNew[i]=arr[elemanSayisi-1-i];

}

**--------ARRAY in elemanlarını sıralama------**

Bir array in tüm elamanlarını yazdırma:

* Arrays.toString() methodunda parametre olarak array in ismini yazarsaniz
* o array in tum elemanlarinı yazdırır.

System.***out***.println(Arrays.*toString*(arr));

Bir Array i sıralama:

* Arrays.sort() methoduna parametre olarak array in ismi yazilirsa

o array in elemanlari kucukten buyuye yazdirir.bu siralamaya kucukten buyuge (Ascending order- natural order) denir.

Hinweis:(natural order)=alfabetik(Stringleri sıralama) and ascending(integerleri kucukten buyuge sıralama)

* BİR NOT

Ascending (artan sıraya göre)

Descending (düşen sıraya göre)

* Char lar sıralanırken java asccii kodları kullanır. bu yuzden buyuk harfler kucuk harflerden önce gelir.

rakamlarin ascii kodunu rakamın degeri gibi dusunur.

ÖRNEK: char lardan olusan bir ascending olusturunuz ve "ascending order" yapiniz

**char** arr[] = {'1','a','A','B','c','\\','%'};

Arrays.*sort*(arr);

System.***out***.println(Arrays.*toString*(arr)); // [%, 1, A, B, \, a, c]

* Dikkat et! Syso nun için yazmadım. Önce sırala sonra yazdır.
* Stringlerden olusan bir array alfabetik olarak sıralanır.
* Boolean icin sort methodu kullanılmaz.

**----------iki Array in eşitliğini kontrol etme---------**

Method: Arrays.equals(arr1,arr2)

**int** arr1[] = {1, 2, 3};

**int** arr2[] = {1, 2, 3};

System.***out***.println(arr1==arr2);//false cunku reference'lar farkli. "==" adresleri de kontrol eder. Iki array farkli iki object oldugundan adresleri farklidir

System.***out***.println(Arrays.*equals*(arr1, arr2));//true verir cunku Arrays.equals() methodu sadece degerleri ve index'leri kontrol eder, adreslere bakmaz.

* Dikkat! Direk syso nun içinde yazdırdım.

**-------bir elemanın Array içindeki varlığını kontrol etme--------**

Method: Arrays.binarySearch(array ismi, aradığımız eleman)

**int** arr3[] = {1,2,3,4,5,6};

System.***out***.println(Arrays.*binarySearch*(arr3, 2));

* Dikkat! Direk syso nun içine yazdım.

1. Bu yöntem aradığımız elemanın index ini verir.
2. Onemli Not: binarySearch() methodunu kullanmadan önce sort() methodunu zorundasiniz. Aksi takdirde; binarySearch() methodu manali bir sonuc vermez.
3. Olmayan bir eleman sorgulatırsak negatif sayilar return eder.negatif olmasi o elemanin arrayde olmadigi anlamina gelir.

Sayi kismi ise o eleman var olsaydi kacinci eleman olurdu anlamindadir.