



# ArrayList

## ArrayList nedir?

ArrayList length'i esnek olan bir Array'dir

## ArrayList'e nicin ihtiyac duyariz?

- Biz array olustururken length'in en basta belirlemek zorundayiz ve daha sonra length'ini degistiremeyiz. Bu durum bizim esnek calismamiza engel olur.
- Bir array'in uzunlugunu degistirmek istedigimizde yeni bir array olusturmamiz gerekir, ArrayList de gerekmez.
- Bir array'den bir eleman silmek istedigimizde yeni bir array olusturmamiz gerekir, ArrayList de gerekmez.



# Array**L**ist

## ArrayList olusturma

```
ArrayList<String> list1 = new ArrayList<String>();
```

```
ArrayList<String> list2 = new ArrayList<>();
```

```
List<String> list3 = new ArrayList<>(); En cok bu kullanilir
```

```
ArrayList<String> list4 = new List<>();
```

**Compile Time Error verir, esitligin sag tarafinda ArrayList kullanmak zorundayiz**

## ArrayList'i nasil yazdiririz?

ArrayList'i ekrana yazdirmek cok kolaydir.

```
System.out.println(list3);
```



## Array**L**ist **M**ethod'leri

### 1) add( )

add() method ArrayList'e eleman eklemek için kullanılır

### Örnek :

```
List<String> hayvan = new ArrayList<>();
```

#### A) add( ) method'u index olmadan çalışabilir

```
hayvan.add("kedi"); // [kedi]
```

```
hayvan.add("yılan"); // [kedi, yılan]
```

#### B) add( ) method'u index ile de çalışabilir

```
hayvan.add(1, "kartal"); // [kedi, kartal, yılan]
```

```
hayvan.add(0, "sinek"); // [sinek, kedi, kartal, yılan]
```

```
hayvan.add(1, "aslan"); // [sinek, aslan, kedi, kartal, yılan]
```

```
System.out.println(hayvan); // [sinek, aslan, kedi, kartal, yılan]
```



## Array**L**ist **M**ethod'lari

### 2) size( )

size( ) method ArrayList'de kac eleman oldugunu gosterir.

Ornek :

```
List<String> hayvan = new ArrayList<>( );
```

```
System.out.println(hayvan.size( )); // 0
```

```
hayvan.add("kedi"); // [kedi]
```

```
hayvan.add("yilan"); // [kedi, yilan]
```

```
System.out.println(hayvan.size( )); // 2
```

### 3) isEmpty( )

isEmpty( ) method'u ArrayList bos ise true, bos degilse false dondurur



## ArrayList Method'ları

### 4) remove()

remove() method'u ArrayList'den belli bir elemani silmek için kullanılır.

**A) remove(index) kullanarak.** Size'dan büyük index yazılırsa exception verir. Index'li remove() methodu ArrayList'de verilen index'deki elemani siler.

```
List<String> hayvan = new ArrayList<>( );  
hayvan.add("kedi"); // [kedi]  
hayvan.add("yılan"); // [kedi, yılan]  
hayvan.remove(1); // index'i 1 olan elemani siler  
System.out.println(hayvan); //[kedi]
```

**NOT:** remove(index) method'u silinen elemani dondurur. Yani method'u **System.out.println()** icinde kullanırsak silinen elemani ekrana yazdırır.

```
System.out.println(hayvan.remove(1)); //yılan
```



## ArrayList Method'ları

**B) remove("eleman")** index'i değil elemanı kullanırsak kullandığımız elemanın ilk kullanıldığı yeri bulur ve siler.

```
List<String> hayvan = new ArrayList<>( );  
    hayvan.add("kedi"); // [kedi]  
    hayvan.add("yılan"); // [kedi, yılan]  
    hayvan.add("kedi"); // [kedi, yılan, kedi]  
    hayvan.remove("kedi");  
    System.out.println(hayvan); // [yılan, kedi]
```

**Not:** Index'siz remove( ) method'u true veya false döndürür.

```
System.out.println(hayvan.remove("kedi")); //true yani kedi eleman olarak vardı ve sildim
```

```
System.out.println(hayvan.remove("tavsan")); // false yani tavsan eleman olarak yoktu  
dolayısıyla silemedim
```



## ArrayL**ist** M**ethod**'lari

### 5) **set( )**

set( ) methodu ArrayList'de var olan bir elemani degistirmeye yarar

```
List<String> hayvan = new ArrayList<>( );
```

```
hayvan.add("kedi"); // [kedi]
```

```
hayvan.add("yilan"); // [kedi, yilan]
```

```
hayvan.set(1, "tavsan");
```

```
System.out.println(hayvan); //[kedi, tavsan]
```

**NOT:** set( ) method'u add( ) method'u yerine kullanilamaz .  
Olmayan bir index ile set( ) kullanilrsa exception verir.

```
hayvan.set(2, "aslan"); // IndexOutOfBoundsException
```



## Array**L**ist **M**ethod'lari

### 6) get(index)

get( ) methodu ArrayList'deki istenen indexdeki elemani dondurur.

```
List<String> hayvan = new ArrayList<>();
```

```
hayvan.add("kedi"); // [kedi]
```

```
hayvan.add("yilan"); // [kedi, yilan]
```

```
System.out.println(hayvan.get(0)); // kedi
```

```
System.out.println(hayvan.get(1)); // yilan
```





## ArrayList Method'lari

### 7) contains()

contains() methodu ArrayList'de bir elemanın var olup olmadığını kontrol eder. Eleman varsa true, yoksa false return eder.

```
List<String> hayvan = new ArrayList<>();
```

```
hayvan.add("kedi"); // [kedi]
```

```
hayvan.add("yılan"); // [kedi, yılan]
```

```
System.out.println(hayvan.contains("kedi")); // true
```

```
System.out.println(hayvan.contains("tavsan")); // false
```



## ArrayL**ist** M**ethod**'lari

8) **Collections.sort()** : sort() methodu ArrayList'deki elemanlari  
kucukten buyuge veya alfabetik siraya gore dizer.

```
List<String> hayvan = new ArrayList<>( );  
hayvan.add("yilan"); // [yilan]  
hayvan.add("kedi"); // [yilan, kedi]  
hayvan.add("tavsan"); // [yilan, kedi, tavsan]  
  
System.out.println(hayvan); //[yilan, kedi, tavsan]  
  
Collections.sort(hayvan);  
System.out.println(hayvan); // [kedi, tavsan, yilan]
```



## ArrayL**ist** Method'lari

### 9) equals( )

equals( ) methodu iki listteki ayni indexteki elemanlari ayni olup olmadigini kontrol eder. Ayni indexteki tum elemanlar ayni ise true return eder, farkli ise false return eder

```
List<String> first = new ArrayList<>( );  
List<String> second = new ArrayList<>( );  
System.out.println(first.equals(second)); // true  
  
first.add("a"); // [a]  
System.out.println(first.equals(second)); // false  
  
second.add("a"); // [a]  
System.out.println(first.equals(second)); // true  
  
first.add("b"); // [a,b]  
second.add(0,"b"); // [b,a]  
System.out.println(first.equals(second)); // false
```



## ArrayL**ist** M**ethod**'lari

### 10) **clear()**

`clear()` methodu ArrayList'teki tum elemanlari siler.  
Return type'i void'dir, hic bir sey donmez

```
List<String> hayvan = new ArrayList<>();  
hayvan.add("yilan"); // [yilan]  
hayvan.add("kedi"); // [yilan, kedi]
```

```
System.out.println(hayvan.isEmpty()); // false  
System.out.println(hayvan.size()); // 2
```

```
hayvan.clear();  
System.out.println(hayvan.isEmpty()); // true  
System.out.println(hayvan.size()); // 0
```



## ArrayList Sorular

- 1) Elemanlari A, C, E, ve F olan bir String ArrayList olusturup ekrana yazdiriniz.
- 2) indexsiz **add()** methodunu kullanarak, B'yi ekleyiniz.  
index'li **add()** methodunu kullanarak, L'yi 1 numarali index'e ekleyiniz.  
ArrayList'i ekrana yazdiriniz, list goyle olmalı; A, L, C, E, F, B.
- 3) **set()** methodu kullanarak, E'yi D yapiniz.  
ArrayList'i ekrana yazdiriniz, list goyle olmalı; A, L, C, D, F, B.
- 4) **remove()** methodu kullanarak, F'yi siliniz.  
ArrayList'i ekrana yazdiriniz, list goyle olmalı; A, L, C, D, B.
- 5) **sort()** methodu kullanarak, elemanlari alfabetik siraya diziniz.  
ArrayList'i ekrana yazdiriniz, list goyle olmalı; A, B, C, D, L.
- 6) **contains()** methodu kullanarak, L'nin list'de var oldugunu ve M'nin list'de var olmadigini dogrulayiniz.
- 7) **size()** methodu kullanarak, list'in kac eleman oldugunu ekrana yazdiriniz.
- 8) **clear()** methodu kullanarak, list'deki tum elemanlari siliniz.
- 9) **isEmpty()** methodu kullanarak, list'deki tum elemanlarin silindigini dogrulayiniz



## Array'i ArrayList'e Cevirmek

```
String [ ] arr = {"tavsan", "serce"};
```

```
List<String> list = Arrays.asList(arr);
```

Uzunlugu degistirilemeyen bir list'e cevirir. Yani; yeni olusturulan listte add(), remove() ve clear() methodlarini kullanamazsiniz. Exception

```
System.out.println(list.size()); // 2
```

```
System.out.println(list); // [tavsan, serce]
```

**NOT:** Eger array'deki bir elemani degistirirseniz list'teki eleman da otomatik olarak degisir. Listteki bir elemani degistirirseniz array de otomatik olarak degisir.

```
list.set(1, "test"); // [tavsan, test]
```

```
arr[0] = "new"; // [new, test]
```

```
System.out.println(Arrays.toString(arr)); // [new, test]
```

```
System.out.println(list); // [new, test]
```



## ArrayList'i Array'e Cevirmek

```
List<String> list = new ArrayList<>();  
list.add("tavsan");  
list.add("horoz");  
System.out.println(list); //[tavsan,horoz]
```

### 1.yontem

```
String arr[ ] = list.toArray(new String[0]);
```

```
System.out.println(arr.length); // 2  
System.out.println(Arrays.toString(arr)); //[tavsan,horoz]
```

### 2.yontem

```
Object arr[ ] = list.toArray( );
```

```
System.out.println(arr.length); // 2  
System.out.println(Arrays.toString(arr)); //[tavsan,horoz]
```