

YMT 112

Algoritma ve Programlama II

Fatih Özkaynak

İçerik

- Değişkenler
- Operatörler
- Kontrol yapıları
- Döngüler
- Metotlar
- Diziler
- Sınıflar

Diziler

- Dizi, bilgisayar belleğinde aynı isim altında genellikle aynı tipten çok sayıda veriyi bir arada saklayan veri yapısıdır.
- **x** adlı bir dizinin mantıksal görünümü yukarıda verilmiştir:
- Dizinin bir adı ve dizi içerisindeki elemanın dizinin kaçınıcı elemanı olduğunu belirten bir **indisi** (subscript, index) vardır.
- Örneğin, **x** adlı dizi için **x** ifadesi bu dizinin 3 numaralı elemanını yani 22 değerini göstermektedir.
- Dizinin indisi Java dilinde daima köşeli parantez ile belirlenir:
- Java dilinde bir dizinin ilk elemanının indisi daima 0'dır; son elemanın indisi ise diziye ayrılan yer sayısının 1 eksiğine eşittir.

4	2	7	22	11	3
0	1	2	3	4	5

Dizi Tanımlanması

- Bir Java programı içinde diziyi tanıtmak için, new sözcüğünü kullanarak belirli bir tipten yeni bir nesne tanımlama yolu izlenir.
- Dizinin tipi, adı ve maksimum eleman sayısı bir bildiri deyimi içinde belirtilir:
 - `Tip DiziAdı[] = new Tip [Eleman Sayısı];`
 - `Tip [] DiziAdı = new Tip [Eleman Sayısı];`
- Örneğin
 - `int [] x= new int [6] ;`

Dizi için ne gereklidir?

- Dizi yapısına, aynı türden bir veri grubunu tümüyle bellekte saklı tutmanın gerekli olduğu uygulamalarda ihtiyaç duyarız.
- Örneğin:
 - Verilerin sıralanması,
 - Bir veri grubuna ait bazı istatistiksel bilgilerin hesaplanması (standart sapma vb.)

```
1  import java.util.Scanner;
2  public class Dizi1{
3  public static void main(String[] args)
4  { Scanner giris = new Scanner(System.in);
5
6  int girilenSayi, toplam, i ;
7  toplam = 0; girilenSayi = 0;
8  for (i = 1; i <= 5; i++)
9  {
10 System.out.print("Bir sayi giriniz: ");
11 girilenSayi = giris.nextInt();
12 toplam = toplam + girilenSayi;
13 }
14 System.out.println(i-1+ " tane sayi girdiniz.");
15 System.out.println("Toplami " +toplam+ " ediyor.");
16 }
17 }
```

```
1  import java.util.Scanner;
2  public class Dizi2{
3  public static void main(String[] args)
4  {
5  Scanner giris = new Scanner(System.in);
6  int sayi[] = new int[6];
7  int toplam, i; toplam = 0;
8  for (i = 1; i <= 5; i++)
9  {
10 System.out.print("Bir sayi giriniz: ");
11 sayi[i] = giris.nextInt();
12 toplam = toplam + sayi[i];
13 }
14 System.out.println("Girdiniz sayilarin toplami: " + toplam);
15 System.out.println("Sayi dizisinin ikinci elemani: " + sayi[2]);
16 }
17 }
```

```
1 public class Dizi5 {
2     public static void main(String[] args){
3         int[] a={2,8,3,7,5};
4         int toplam;
5         toplam=topla_dizi(a,5);
6         System.out.println("Toplam: "+toplam);
7     }
8
9     static int topla_dizi(int [] x,int n){
10         int i, ictoplam;
11         ictoplam=0;
12         for(i=0; i<n; i++) {
13             ictoplam+=x[i];
14         }
15         return ictoplam;
16     }
17 }
```


Tek Boyutlu Bir Dizi

6	34	23	-8	123	87	19	12
---	----	----	----	-----	----	----	----

↑
indeks [5]

2 Boyutlu(Düzenli) Bir Dizi

6	34	23	-8	123	87	19	12
2	43	3	22	-73	78	17	22
16	4	17	63	7	-31	-18	32
0	125	33	99	981	31	16	0

↓
index [0][5]

←
indeks [1][6]

↑
indeks [3][2]

2 Boyutlu(Düzensiz) Bir Dizi

6	34	23	-8	123	87	19	12			
2	43	3	22	-73	78	17	22	9	255	127
16	4	17	63	7	-31	-18	32	8		
0	125	33	99	981	31	16				

Diagram illustrating a 2D array (Düzensiz) with 4 rows and 11 columns. The array is shown with its elements and indices:

- Row 0: 6, 34, 23, -8, 123, 87, 19, 12
- Row 1: 2, 43, 3, 22, -73, 78, 17, 22, 9, 255, 127
- Row 2: 16, 4, 17, 63, 7, -31, -18, 32, 8
- Row 3: 0, 125, 33, 99, 981, 31, 16

Indices and corresponding values are indicated by arrows:

- `indeks[0][5]` points to the value 87 in Row 0, Column 5.
- `indeks[1][9]` points to the value 255 in Row 1, Column 9.
- `indeks[2][7]` points to the value 32 in Row 2, Column 7.
- `indeks[3][2]` points to the value 33 in Row 3, Column 2.

```
1 public class CokBoyutluDizi5 {
2     public static void main(String[] args){
3         double[][] dizi=new double[3][3];
4         int i,j,sayimiz;
5         sayimiz=2;
6         for(i=0; i<=2; i++) {
7             for(j=0; j<=2; j++) {
8                 dizi[i][j]=faktoriyel(sayimiz);
9                 System.out.println(sayimiz+"!="+dizi[i][j]);
10                sayimiz=sayimiz+2;
11            }
12        }
13    }
14
15    static double faktoriyel(int n){
16        double fakt;
17        int i;
18        fakt=1.0;
19        for(i=1; i<=n; i++) {
20            fakt=fakt*i;
21        }
22        return fakt;
23    }
24 }
```

Dizi Sinifi (class array):

Array sinifi, **java.util** paketindedir. Dizi islemleri için bir çok metot barındırır. Bunlardan bir kaçı şunlardır:

sort()	Diziyi tasnif etmek, alfabetik siraya koymak için kullanılır. Ör: <i>Arrays.sort(a);</i>
binarySearch()	Tasnif edilmiş bir dizide bir deger aramak için kullanılır.
equals()	Dizileri karsilastirmek için kullanılır. Ör: <i>Arrays.equals(birinciDizi[], ikinciDizi[]); //true-false</i>
fill()	Dizinin içini doldurmak için kullanılır. Ör: <i>Arrays.fill(aDizisi, 0);</i>

```
public class Program {  
    public static void main(String[] args) {  
  
        // Create int array.  
        int[] array = new int[5];  
  
        // Assign first three elements.  
        array[0] = 1;  
        array[1] = 10;  
        array[2] = 100;  
  
        // Loop over elements.  
        for (int i = 0; i < array.length; i++) {  
            // Get value.  
            int value = array[i];  
            // Print value.  
            System.out.println(value);  
        }  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // Create an array of four chars.  
        char[] values = new char[4];  
        values[0] = 'j';  
        values[1] = 'a';  
        values[2] = 'v';  
        values[3] = 'a';  
  
        // Loop over array with for-loop.  
        for (char value : values) {  
            System.out.println(value);  
        }  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // Two input arrays.  
        int[] array1 = {1, 3, 5};  
        String[] array2 = {"frog", "toad", "squirrel"};  
  
        // Array lengths.  
        System.out.println(array1.length);  
        System.out.println(array2.length);  
  
        // First elements in each array.  
        System.out.println(array1[0]);  
        System.out.println(array2[0]);  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        boolean[] values = { false, true, true, true };  
  
        // Loop over array elements in reverse order.  
        for (int i = values.length - 1; i >= 0; i--) {  
            System.out.println(values[i]);  
        }  
    }  
}
```



```
public class Program {  
    public static void main(String[] args) {  
  
        // Two empty arrays.  
        int[] array1 = {};  
        int[] array2 = new int[0];  
  
        // Each has zero length.  
        System.out.println(array1.length);  
        System.out.println(array2.length);  
    }  
}
```

```
import java.util.Arrays;

public class Program {
    public static void main(String[] args) {

        int[] array = { 100, 20, 0, 200 };

        // Call Arrays.sort on the int array.
        Arrays.sort(array);

        for (int elem : array) {
            System.out.println(elem);
        }
    }
}
```

```
import java.util.Arrays;

public class Program {
    public static void main(String[] args) {

        int[] values = new int[10];

        // Fill array with this number.
        Arrays.fill(values, 5);

        for (int value : values) {
            System.out.print(value);
            System.out.print(' ');
        }
    }
}
```

```
import java.util.Arrays;

public class Program {
    public static void main(String[] args) {

        int[] values = { 10, 20, 30, 40 };

        // Copy and display first 3 elements.
        int[] copy = Arrays.copyOf(values, 3);
        for (int value : copy) {
            System.out.println(value);
        }
        System.out.println();

        // Copy five elements.
        int[] copy2 = Arrays.copyOf(values, 5);
        for (int value : copy2) {
            System.out.println(value);
        }
    }
}
```

```
import java.util.Arrays;

public class Program {
    public static void main(String[] args) {

        int[] values = { 0, 10, 20, 30, 40, 50 };

        // Copy elements from index 2 to 5 (2, 3, 4).
        int[] result = Arrays.copyOfRange(values, 2, 5);
        for (int value : result) {
            System.out.println(value);
        }
    }
}
```

```
public class Program {  
  
    static void scan(int[] buffer) {  
        for (int i = 0; i < buffer.length; i++) {  
            // Terminate loop when sentinel element is detected.  
            if (buffer[i] == -1) {  
                break;  
            }  
            System.out.println(buffer[i]);  
        }  
        System.out.println();  
    }  
  
    public static void main(String[] args) {  
  
        // The sentinel element is the sixth one.  
        int[] buffer = { 10, 20, 30, 25, 35, -1, 50, 55 };  
        scan(buffer);  
        // Make the third element the sentinel.  
        buffer[2] = -1;  
        scan(buffer);  
    }  
}
```

```
import java.util.Arrays;

public class Program {
    public static void main(String[] args) {

        int[] values = { 505, 100, 605 };
        // Call Arrays.toString to display the elements.
        System.out.println(Arrays.toString(values));
    }
}
```

```
import java.util.Arrays;

public class Program {
    public static void main(String[] args) {

        int[] values = { 10, 20, 30 };
        int[] values2 = { 100, 200, 300 };

        // Merge the two arrays with for-loops.
        int[] merge = new int[values.length + values2.length];
        for (int i = 0; i < values.length; i++) {
            merge[i] = values[i];
        }
        for (int i = 0; i < values2.length; i++) {
            merge[i + values.length] = values2[i];
        }

        // Display the merged array.
        System.out.println(Arrays.toString(merge));
    }
}
```



```
public class Program {  
    public static void main(String[] args) {  
  
        // Create a char array of 26 characters.  
        // ... Add all letters to it.  
        char[] array = new char[26];  
        int index = 0;  
        for (char c = 'a'; c <= 'z'; c++) {  
            array[index++] = c;  
        }  
        String result = new String(array); // Convert to a string.  
        // ... Display parts of our new string.  
        System.out.println(result.startsWith("abc"));  
        System.out.println(result.length());  
        System.out.println(result);  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // Use array initializer syntax.  
        char[] array = { 'a', 'r', 't', 'i', 's', 't' };  
        // Add only the first three characters to a new string.  
        String result = new String(array, 0, 3);  
        System.out.println(result);  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // Numbers can be stored in a char array.  
        // ... These indicate a char based on ASCII.  
        char[] values = new char[3];  
        values[0] = 97;  
        values[1] = 98;  
        values[2] = 99;  
        System.out.println(values);  
  
        // We can specify letters as well.  
        char[] values2 = { 'a', 'b', 'c' };  
        System.out.println(values2);  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        String value = "cat";  
  
        // Convert string to a char array.  
        char[] array = value.toCharArray();  
        array[0] = 'h';  
  
        // Loop over chars in the array.  
        for (char c : array) {  
            System.out.println(c);  
        }  
    }  
}
```

```
import java.lang.StringBuilder;

public class Program {
    public static void main(String[] args) {
```

```
        long t1 = System.currentTimeMillis();
```

```
        // Version 1: create string from a char array.
```

```
        for (int i = 0; i < 1000000; i++) {
            char[] array = new char[10];
            for (int v = 0; v < 10; v++) {
                array[v] = '?';
            }
            String result = new String(array);
        }
```

```
        long t2 = System.currentTimeMillis();
```

```
        // Version 2: create string from a StringBuilder.
```

```
        for (int i = 0; i < 1000000; i++) {
            StringBuilder builder = new StringBuilder();
            for (int v = 0; v < 10; v++) {
                builder.append('?');
            }
            String result = builder.toString();
        }
```

```
        long t3 = System.currentTimeMillis();
```

```
        // ... Benchmark timings.
```

```
        System.out.println(t2 - t1);
```

```
        System.out.println(t3 - t2);
```

```
    }
```

```
}
```

38 ms:

char[] array

81 ms:

StringBuilder append

```
public class Program {  
    public static void main(String[] args) {  
  
        // Create int array with four elements.  
        int[] values = { 10, 20, 30, 40 };  
        // ... Loop over the array's elements.  
        for (int value : values) {  
            System.out.println(value);  
        }  
  
        // Create int array with three elements in separate statements.  
        int[] alternative = new int[3];  
        alternative[0] = 100;  
        alternative[1] = 200;  
        alternative[2] = -100;  
        // ... Use for-loop to access all elements.  
        for (int i = 0; i < alternative.length; i++) {  
            System.out.println(alternative[i]);  
        }  
    }  
}
```

```
public class Program {  
  
    static int[] getEmployees() {  
        // Create an int array and return it.  
        int[] array = new int[6];  
        array[0] = 9;  
        array[1] = 11;  
        array[2] = 15;  
        array[3] = 19;  
        array[4] = 29;  
        array[5] = 55;  
        return array;  
    }  
  
    public static void main(String[] args) {  
  
        // Loop over an array returned by a method.  
        for (int e : getEmployees()) {  
            System.out.println(e);  
        }  
    }  
}
```

```
class Building {  
    int[] apartmentIds = new int[10];  
  
    public void addResidentAt(int id) {  
        // Add to the array at this index.  
        apartmentIds[id]++;  
    }  
  
    public void removeResidentAt(int id) {  
        // Subtract from element value.  
        apartmentIds[id]--;  
    }  
  
    public int getOccupancyAt(int id) {  
        // Return element value.  
        return apartmentIds[id];  
    }  
}
```


Output

```
public class Program {  
    public static void main(String[] args) {  
  
        // Create a Building.  
        // ... Some residents move in and one leaves.  
        Building b = new Building();  
        b.addResidentAt(5);  
        b.addResidentAt(3);  
        b.addResidentAt(9);  
        b.removeResidentAt(5);  
  
        // Display occupancy of apartments.  
        for (int i = 0; i < 10; i++) {  
            System.out  
                .println(Integer.toString(i) + ": " + b.getOccupancyAt(i));  
        }  
    }  
}
```

0: 0
1: 0
2: 0
3: 1
4: 0
5: 0
6: 0
7: 0
8: 0
9: 1

```
public class Program {  
    public static void main(String[] args) {  
        int[] array = null;  
        // We must first check for null before looping over an array.  
        // ... This causes a runtime error.  
        for (int value : array) {  
            System.out.println(value);  
        }  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // This array has just 5 elements.  
        // ... So the only valid indexes are 0, 1, 2, 3 and 4.  
        int[] array = { 40, 50, -60, -70, 80 };  
  
        // This causes an exception.  
        array[10] = 1000;  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        int[] items = { 10, 20, 40 };  
  
        // Clone the int array.  
        // ... All the elements are copied into a new array.  
        int[] copy = items.clone();  
  
        // When the copy is modified, the original "items" is not affected.  
        copy[0] = -100;  
  
        System.out.println(items[0]);  
        System.out.println(copy[0]);  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // Add different objects to an Object array.  
        Object[] elements = new Object[4];  
        elements[0] = "cat";  
        elements[1] = 100;  
        elements[2] = new StringBuilder("abc");  
        elements[3] = 1.2;  
  
        // Print the objects in a for-loop.  
        for (Object e : elements) {  
            System.out.println(e);  
        }  
    }  
}
```

```
public class Program {  
  
    static void display(Object[] array) {  
        for (Object v : array) {  
            // Get the class object for the element.  
            Class<? extends Object> c = v.getClass();  
            // Test the class against known classes.  
            if (c == String.class) {  
                System.out.println("Found String: " + v);  
            } else if (c == Integer.class) {  
                System.out.println("Found Integer: " + v);  
            } else if (c == StringBuilder.class) {  
                System.out.println("Found StringBuilder: " + v);  
            } else if (c == Double.class) {  
                System.out.println("Found Double: " + v);  
            }  
        }  
    }  
}
```

```
public static void main(String[] args) {  
    Object[] elements = new Object[4];  
    elements[0] = "spark";  
    elements[1] = 500;  
    elements[2] = new StringBuilder("therapeutics");  
    elements[3] = 63.5;  
  
    // Pass our object array to the display method.  
    display(elements);  
}  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        Object[] elements = new Object[2];  
        elements[0] = "cat";  
        elements[1] = "bird";  
        // This statement causes an error.  
        String[] values = (String[]) elements;  
        for (String v : values) {  
            System.out.println(v);  
        }  
    }  
}
```



```
public class Program {  
    public static void main(String[] args) {  
  
        String[] codes = new String[2];  
        codes[0] = "ABC10";  
        codes[1] = "DEF20";  
        // We can cast a String array to an Object array safely.  
        Object[] values = (Object[]) codes;  
        for (Object v : values) {  
            System.out.println(v);  
        }  
    }  
}
```

```
import java.util.ArrayList;

public class Program {
    public static void main(String[] args) {

        // Create an ArrayList of strings.
        ArrayList<String> list = new ArrayList<>();
        list.add("cat");
        list.add("box");
        list.add("elephant");

        // Use toArray to copy ArrayList to string array.
        String[] array = new String[list.size()];
        array = list.toArray(array);

        // Loop over the string elements in the array.
        for (String item : array) {
            System.out.println(item);
        }
    }
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // This string array contains three values.  
        String[] values = { "cat", "dog", "spider" };  
  
        // Combine these strings together.  
        String joined = String.join("", values);  
        System.out.println(joined);  
  
        // Separate the combined string with split.  
        String[] values2 = joined.split("");  
        for (String v : values2) {  
            System.out.println(v);  
        }  
    }  
}
```

```
import java.util.Arrays;

public class Program {
    public static void main(String[] args) {

        // Create a string array with four values.
        String[] values = new String[4];
        values[0] = "zoo";
        values[1] = "marina";
        values[2] = "amphitheatre";
        values[3] = "colloseum";

        // Sort the strings.
        Arrays.sort(values);

        // Display the sorted strings.
        for (String v : values) {
            System.out.println(v);
        }
    }
}
```

```
import java.util.Arrays;
import java.util.OptionalInt;
import java.util.stream.IntStream;

public class Program {
    public static void main(String[] args) {

        int[] array = { 10, 20, 30, 40, 50, 60 };
        // Convert array to Stream.
        IntStream stream = Arrays.stream(array);

        // Filter out values less than 40.
        OptionalInt result = stream.filter(value -> value >= 40)
            .findFirst();
        // If a result is present, display it as an int.
        if (result.isPresent()) {
            // This is the first value returned by the filter.
            System.out.println(result.getAsInt());
        }
    }
}
```

```
import java.util.Arrays;
import java.util.OptionalInt;
import java.util.stream.IntStream;

public class Program {
    public static void main(String[] args) {

        int[] array = { -100, -200 };
        IntStream stream = Arrays.stream(array);

        // This filters out all elements.
        OptionalInt result = stream.filter(value -> value >= 0)
            .findFirst();
        System.out.println(result.getAsInt());
    }
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // Create new ArrayList.  
        ArrayList<Integer> elements = new ArrayList<>();  
  
        // Add three elements.  
        elements.add(10);  
        elements.add(15);  
        elements.add(20);  
  
        // Get size and display.  
        int count = elements.size();  
        System.out.println("Count: " + count);  
  
        // Loop through elements.  
        for (int i = 0; i < elements.size(); i++) {  
            int value = elements.get(i);  
            System.out.println("Element: " + value);  
        }  
    }  
}
```

```
import java.util.ArrayList;

public class Program {
    public static void main(String[] args) {

        // Create ArrayList and add three Integers.
        ArrayList<Integer> list = new ArrayList<>();
        list.add(10);
        list.add(20);
        list.add(30);

        // Display values.
        for (int value : list) {
            System.out.println(value);
        }
    }
}
```



```
public class Program {  
    public static void main(String[] args) {  
  
        ArrayList<Integer> list = new ArrayList<>();  
        list.add(101);  
        list.add(100);  
        list.add(99);  
        list.add(100);  
        list.add(99);  
  
        // Search for values.  
        int index = list.indexOf(100);  
        int lastIndex = list.lastIndexOf(100);  
        int notFound = list.indexOf(200);  
  
        // Display results.  
        System.out.println(index);      // 1  
        System.out.println(lastIndex); // 3  
        System.out.println(notFound);   // -1  
    }  
}
```

```
import java.util.ArrayList;

public class Program {
    public static void main(String[] args) {

        // Create an ArrayList.
        ArrayList<String> list = new ArrayList<>();
        list.add("Venus"); // [0]
        list.add("Mars"); // [1]
        list.add("Earth"); // [2]

        // Set index 0 to a new String.
        list.set(0, "Saturn");

        for (String value : list) {
            System.out.println(value);
        }
    }
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // Create a new ArrayList.  
        ArrayList<Integer> list = new ArrayList<>();  
  
        // Add one element.  
        list.add(1);  
  
        // This prints false: it is not empty.  
        System.out.println(list.isEmpty());  
  
        // Clear ArrayList.  
        list.clear();  
  
        // It is now empty.  
        if (list.isEmpty()) {  
            System.out.println("Empty");  
        }  
    }  
}
```

```
import java.util.ArrayList;
import java.util.List;
```

Output

```
public class Program {
    public static void main(String[] args) {

        // ... Add five integers to an ArrayList.
        ArrayList<Integer> list = new ArrayList<>();
        list.add(5);
        list.add(10);
        list.add(15);
        list.add(20);
        list.add(25);

        // ... Get sub list from 1 to 3.
        List<Integer> sub = list.subList(1, 3);

        // ... Display sub list.
        for (int value : sub) {
            System.out.println(value);
        }

        // ... Set the first element in "sub" to -1.
        // This is reflected in the original ArrayList.
        sub.set(0, -1);
        System.out.println(list.get(1));
    }
}
```

10
15
-1

```
import java.util.ArrayList;
import java.util.List;

public class Program {
    public static void main(String[] args) {

        ArrayList<Integer> list = new ArrayList<>();
        list.add(7);
        list.add(8);
        list.add(9);

        // Create an empty array and pass to toArray.
        Integer[] array = {};
        array = list.toArray(array);

        // Our array now has the ArrayList's elements.
        for (int elem : array) {
            System.out.println(elem);
        }
    }
}
```

```
import java.util.ArrayList;

public class Program {

    static void addCats(ArrayList<String> list) {
        list.add("Fluffy");
        list.add("Max");
    }

    public static void main(String[] args) {

        ArrayList<String> list = new ArrayList<>();

        // Call method and pass ArrayList as argument.
        addCats(list);

        for (String value : list) {
            System.out.println(value);
        }
    }
}
```

```
class Philosopher {
    public int value;
    public String name;

    public Philosopher(int value, String name) {
        this.value = value;
        this.name = name;
    }

    public String toString() {
        return "value = " + this.value + ", name = " + this.name;
    }
}

public class Program {
    public static void main(String[] args) {

        // Create an ArrayList of objects.
        ArrayList<Philosopher> list = new ArrayList<>();
        list.add(new Philosopher(1, "Socrates"));
        list.add(new Philosopher(2, "Plato"));

        // Display our objects.
        for (Philosopher p : list) {
            System.out.println(p);
        }
    }
}
```

```
import java.util.ArrayList;
import java.util.Collections;

public class Program {
    public static void main(String[] args) {

        // Create ArrayList.
        ArrayList<Integer> list = new ArrayList<>();
        list.add(10);
        list.add(1);
        list.add(100);
        list.add(5);

        // Min and max.
        int minimum = Collections.min(list);
        int maximum = Collections.max(list);

        System.out.println(minimum);
        System.out.println(maximum);
    }
}
```



```
import java.util.Collections;
import java.util.ArrayList;

public class Program {
    public static void main(String[] args) {

        ArrayList<String> list = new ArrayList<>();
        list.add("cat");
        list.add("bird");
        list.add("ant");
        list.add("dog");

        // Sort the elements alphabetically.
        Collections.sort(list);

        for (String value : list) {
            System.out.println(value);
        }
    }
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        ArrayList<Integer> values = new ArrayList<>();  
        Integer[] array = { 10, 20, 30 };  
  
        // Add all elements in array to ArrayList.  
        Collections.addAll(values, array);  
  
        // Display.  
        for (int value : values) {  
            System.out.print(value);  
            System.out.print(" ");  
        }  
        System.out.println();  
  
        // Add more elements.  
        Collections.addAll(values, 40, 50);  
  
        // Display.  
        for (int value : values) {  
            System.out.print(value);  
            System.out.print(" ");  
        }  
    }  
}
```

```
import java.util.ArrayList;

public class Program {

    static ArrayList<Integer> values = new ArrayList<>();

    public static void main(String[] args) {

        // Use static ArrayList.
        values.add(10);
        values.add(100);
        System.out.println(values.size());

        // Use static ArrayList in another method.
        addMore();
        System.out.println(values.size());

    }

    static void addMore() {
        values.add(1000);
    }

}
```

```
import java.util.Vector;

public class Program {
    public static void main(String[] args) {

        // Add two values to Vector.
        Vector<Integer> v = new Vector<>();
        v.add(100);
        v.add(1000);

        // Display values.
        for (int value : v) {
            System.out.println(value);
        }
    }
}
```

```
import java.util.Vector;

public class Program {
    public static void main(String[] args) {

        Vector<Integer> v = new Vector<>();
        v.add(20);
        v.add(30);
        v.add(40);

        // ... Get element 0.
        System.out.println(v.get(0));

        // ... Loop over all indexes, calling get.
        for (int i = 0; i < v.size(); i++) {
            int value = v.get(i);
            System.out.println(value);
        }
    }
}
```

```
import java.util.Vector;

public class Program {
    public static void main(String[] args) {

        // Create one-element Vector.
        Vector<Integer> v = new Vector<>();
        v.add(20);

        // Set index 0.
        int old = v.set(0, 30);

        // Display old and current value.
        System.out.println(old);
        System.out.println(v.get(0));

        // Set index 0 with setElementAt.
        v.setElementAt(40, 0);
        System.out.println(v.get(0));
    }
}
```

```
import java.util.ArrayList;
import java.util.Vector;

public class Program {
    public static void main(String[] args) {

        // Create new ArrayList.
        ArrayList<String> values = new ArrayList<>();
        values.add("cat");
        values.add("dog");
        values.add("mouse");

        // Add ArrayList to Vector.
        Vector<String> v = new Vector<String>();
        v.addAll(values);

        // Display size of Vector.
        System.out.println(v.size());
    }
}
```

```
import java.util.Vector;

public class Program {
    public static void main(String[] args) {

        // Add ArrayList to Vector.
        Vector<String> v = new Vector<>();
        v.add("one");
        v.add("two");
        v.add("three");

        // ... Convert Vector into String array.
        String[] array = {};
        String[] result = v.toArray(array);
        for (String value : result) {
            System.out.println(value);
        }
    }
}
```



```
import java.util.Vector;

public class Program {
    public static void main(String[] args) {

        Vector<Integer> v = new Vector<>();
        v.add(5);
        v.add(10);
        v.add(15);
        v.add(20);

        // See if 15 is present.
        if (v.contains(15)) {
            System.out.println("Contains 15");
        }

        // Get index of value 20.
        int index = v.indexOf(20);
        System.out.println("index of 20 = " + index);
    }
}
```

```
import java.util.ArrayList;
import java.util.Vector;

public class Program {
    public static void main(String[] args) {

        long t1 = System.currentTimeMillis();

        // ... Add to ArrayList.
        ArrayList<Integer> a = new ArrayList<>();
        for (int i = 0; i < 100000; i++) {
            a.add(i);
        }

        long t2 = System.currentTimeMillis();

        // ... Add to Vector.
        Vector<Integer> v = new Vector<>();
        for (int i = 0; i < 100000; i++) {
            v.add(i);
        }

        long t3 = System.currentTimeMillis();

        // ... Times.
        System.out.println(t2 - t1);
        System.out.println(t3 - t2);
    }
}
```

Results

5 ms, ArrayList
10 ms, Vector

```
import java.util.ArrayList;
import java.util.HashSet;

public class Program {

    static ArrayList<String> removeDuplicates(ArrayList<String> list) {

        // Store unique items in result.
        ArrayList<String> result = new ArrayList<>();

        // Record encountered Strings in HashSet.
        HashSet<String> set = new HashSet<>();

        // Loop over argument list.
        for (String item : list) {

            // If String is not in set, add it to the list and the set.
            if (!set.contains(item)) {
                result.add(item);
                set.add(item);
            }
        }
        return result;
    }
}
```

```
public static void main(String[] args) {  
  
    ArrayList<String> list = new ArrayList<>();  
    list.add("dog");  
    list.add("cat");  
    list.add("dog");  
    list.add("dog");  
    list.add("cat");  
    list.add("bird");  
  
    // Remove duplicates from ArrayList of Strings.  
    ArrayList<String> unique = removeDuplicates(list);  
    for (String element : unique) {  
        System.out.println(element);  
    }  
}  
}
```

```
import java.util.ArrayList;

public class Program {

    public static void main(String[] args) {

        // Append three Strings to the ArrayList.
        ArrayList<String> list = new ArrayList<>();
        list.add("abc");
        list.add("DEF");
        list.add("ghi");

        // Join with an empty delimiter to concat all strings.
        String result = String.join("", list);
        System.out.println(result);
    }
}
```

```
import java.util.ArrayList;

public class Program {

    static String convertToString(ArrayList<Integer> numbers) {
        StringBuilder builder = new StringBuilder();
        // Append all Integers in StringBuilder to the StringBuilder.
        for (int number : numbers) {
            builder.append(number);
            builder.append(":");
        }
        // Remove last delimiter with setLength.
        builder.setLength(builder.length() - 1);
        return builder.toString();
    }

    public static void main(String[] args) {
        // Create an ArrayList of three ints.
        ArrayList<Integer> numbers = new ArrayList<>();
        numbers.add(10);
        numbers.add(200);
        numbers.add(3000);

        // Call conversion method.
        String result = convertToString(numbers);
        System.out.println(result);
    }
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // Create 2-dimensional array.  
        int[][] values = new int[4][4];  
  
        // Assign three elements in it.  
        values[0][0] = 1;  
        values[1][1] = 2;  
        values[3][2] = 3;  
  
        // Loop over top-level arrays.  
        for (int i = 0; i < values.length; i++) {  
  
            // Loop and display sub-arrays.  
            int[] sub = values[i];  
            for (int x = 0; x < sub.length; x++) {  
                System.out.print(sub[x] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

```
public class Program {  
    public static void main(String[] args) {  
  
        // Create an array of String arrays: a jagged array.  
        String[][] values = new String[2][];  
  
        // Fill first row with 2-element array.  
        values[0] = new String[2];  
        values[0][0] = "cat";  
        values[0][1] = "dog";  
  
        // Use 3-element array for second row.  
        values[1] = new String[3];  
        values[1][0] = "fish";  
        values[1][1] = "bird";  
        values[1][2] = "lizard";  
  
        // Display rows and elements.  
        for (String[] array : values) {  
            for (String element : array) {  
                System.out.print(element);  
                System.out.print(" ");  
            }  
            System.out.println();  
        }  
    }  
}
```