Java Arrays

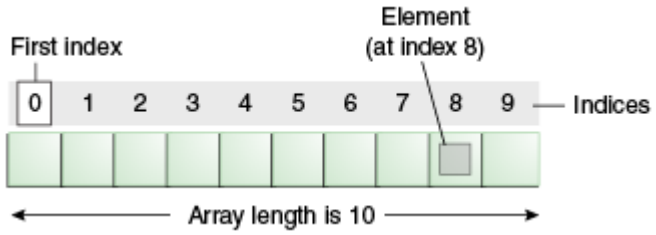
Normally, an array is a collection of similar type of elements which has contiguous memory location.**Java array** is an object which contains elements of a similar data type. Additionally, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.

Unlike C/C++, we can get the length of the array using the length member. In C/C++, we need to use the sizeof operator.

In Java, array is an object of a dynamically generated class. Java array inherits the Object class, and implements the Serializable as well as Cloneable interfaces. We can store primitive values or objects in an array in Java. Like C/C++, we can also create single dimentional or multidimentional arrays in Java.

Moreover, Java provides the feature of anonymous arrays which is not available in C/C++.



Advantages

- **Code Optimization:** It makes the code optimized, we can retrieve or sort the data efficiently.
- **Random access:** We can get any data located at an index position.

Disadvantages

- **Size Limit:** We can store only the fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in Java which grows automatically.

Types of Arrays in java

There are two types of arrays.

- Single Dimensional Array
- Multidimensional Array

Single Dimensional Array in Java

Syntax to Declare an Array in Java

1. dataType[] arr; (or)
2. dataType []arr; (or)
3. dataType arr[];

Instantiation of an Array in Java

1. arrayRefVar=**new** datatype[size];

Example: (Method 1)

```
int a[]=new int[5];//declaration and instantiation

a[0]=10;//initialization
a[1]=20;
a[2]=70;
a[3]=40;
a[4]=50;

//traversing array
for(int i=0;i<a.length;i++)//length is the property of array
{
System.out.println(a[i]);
}
```

```
10
20
70
40
50
```

Example: (Method 2)

```
int a[]={33,3,4,5};//declaration, instantiation and initialization

//printing array

for(int i=0;i<a.length;i++)//length is the property of array

{

System.out.println(a[i]);  }
```

```
33
3
4
5
```

For-each Loop for Java Array

We can also print the Java array using **for-each loop.** The Java for-each loop prints the array elements one by one. It holds an array element in a variable, then executes the body of the loop.

The syntax of the for-each loop is given below:

```
for(data_type variable:array)
{
//body of the loop
}
```

Example:

```
int arr[]={33,34,35};

//printing array using for-each loop

for(int i:arr)
{
System.out.println(i);
}
```

```
33
34
35
```

Anonymous Array in Java

Java supports the feature of an anonymous array, so you don't need to declare the array while passing an array to the method.

Example:

```
//Java Program to demonstrate the way of passing an anonymous array to method.

public class Array3
{

//creating a method which receives an array as a parameter

static void printArray(int arr[])
{
for(int i=0;i<arr.length;i++)
{
System.out.println(arr[i]);
}
}
```

```
public static void main(String args[])
{
printArray(new int[]{10,22,44,66});//passing anonymous array to method
}

}
```

```
10
22
44
66
```

Multidimensional Array in Java

In such case, data is stored in row and column based index (also known as matrix form).

Syntax to Declare Multidimensional Array in Java

1. dataType[][] arrayRefVar; (or)
2. dataType [][]arrayRefVar; (or)
3. dataType arrayRefVar[][]; (or)
4. dataType []arrayRefVar[];

Example to instantiate Multidimensional Array in Java

1. `int[][] arr=new int[3][3];//3 row and 3 column`

Example to initialize Multidimensional Array in Java

1. arr[0][0]=1;
2. arr[0][1]=2;
3. arr[0][2]=3;
4. arr[1][0]=4;
5. arr[1][1]=5;
6. arr[1][2]=6;
7. arr[2][0]=7;
8. arr[2][1]=8;
9. arr[2][2]=9;

Example:

```
//Java Program to illustrate the use of multidimensional array

public class Array5
{

public static void main(String args[])
```

```
{

//declaring and initializing 2D array
int arr[][]={{1,2,4},{2,4,6},{4,4,7}};

//printing 2D array
for(int i=0;i<3;i++)
{
 for(int j=0;j<3;j++)
{
   System.out.print(arr[i][j] + " ");
 }
 System.out.println();
}

}
}
```

```
1 2 4
2 4 6
4 4 7
```

Jagged Array in Java

If we are creating odd number of columns in a 2D array, it is known as a jagged array. In other words, it is an array of arrays with different number of columns.

Example:

```
//Java Program to illustrate the jagged array

public class Array6
{

public static void main(String[] args)
{
        //declaring a 2D array with odd columns
        int arr[][] = new int[3][];
        arr[0] = new int[3];
        arr[1] = new int[4];
        arr[2] = new int[2];

        //initializing a jagged array
        int count = 0;

        for (int i=0; i<arr.length; i++)
            {
        for(int j=0; j<arr[i].length; j++)
            {
                arr[i][j] = count++;
            }
            }
```

```
        //printing the data of a jagged array
        for (int i=0; i<arr.length; i++)
           {
        for (int j=0; j<arr[i].length; j++)
           {
              System.out.print(arr[i][j]+" ");
        }
           System.out.println();//new line
        }
    }
}
```

Example: Java Program to demonstrate the way of passing an array to method. And find minimum number from it

```
class Array2
{

//creating a method which receives an array as a parameter

static void min(int arr[])
{

int temp=arr[0];

for(int i=1;i<arr.length;i++)
{
 if(temp>arr[i])
{
  temp=arr[i];
}
}
System.out.println(temp);

}

public static void main(String args[])
{
int a[]={10,05,20,02};//declaring and initializing an array

min(a);//passing array to method
}}
```

Example: Java Program to return an array from the method

```
class Array4
{

//creating method which returns an array

static int[] get()
{
return new int[]{10,30,50,90,60};
}

public static void main(String args[])
{

//calling method which returns an array
int arr[]=get();

//printing the values of an array

for(int i=0;i<arr.length;i++)
{
System.out.println(arr[i]);
}
}
}
```

```
10
30
50
90
60
```

Example: Java Program to demonstrate the case of ArrayIndexOutOfBoundsException in a Java Array.

```
public static void main(String args[])
{
int arr[]={50,60,70,80};

for(int i=0;i<=arr.length;i++)
{
System.out.println(arr[i]);
}
}
```

```
50
60
70
80
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4
        at Array4.main(Array4.java:37)
```