Subject: Java Programming
Subject Code: 28541

TANJILA TASKIN
Instructor(Tech)Computer
Dhaka Mohila Polytechnic Institute

Chapter:5

Array

5.1 Array in java

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

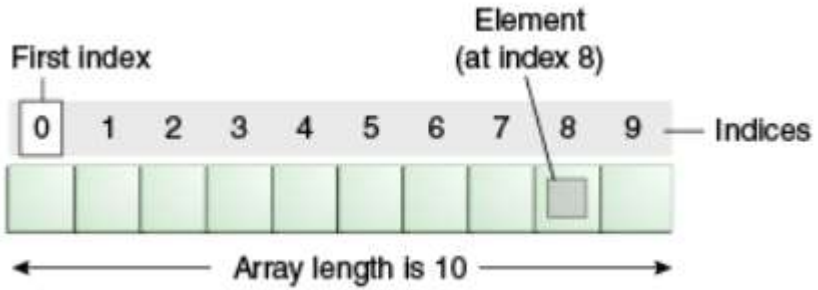
To declare an array, define the variable type with **square brackets**:

```
String[] cars;
```

We have now declared a variable that holds an array of strings. To insert values to it, we can place the values in a comma-separated list, inside curly braces:

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.

- We can store primitive values or objects in an array in Java.



- The index is an integer, and its value is between [0, length of the Array - 1]. For example an Array to hold 5 elements has indices 0, 1, 2, 3, and 4.

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};


public class Main {
  public static void main(String[] args) {
    String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
    for (int i = 0; i < cars.length; i++) {
      System.out.println(cars[i]);
    }
  }
}
```

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.
- **Manual indexing**

Types of Array in java

There are two types of array:

- Single Dimensional Array
- Multidimensional Array
    - Two-dimensional array
    - Three-dimensional array

Single Dimensional Array in Java

Syntax:

- dataType[] arr; (or)
- dataType arr[];

Instantiation of an Array in Java

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

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

Declaration, Instantiation and Initialization of Java Array

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


class Testarray1{
public static void main(String args[]){
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]);
}}
```

```
//Java Program to illustrate how to declare, instantiate, initialize
class Testarray{
public static void main(String args[]){
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]);
}}
```

Multidimensional arrays

An array with more than one dimension is known as a multi-dimensional array. The most commonly used multi-dimensional arrays are 2-D and 3-D arrays.

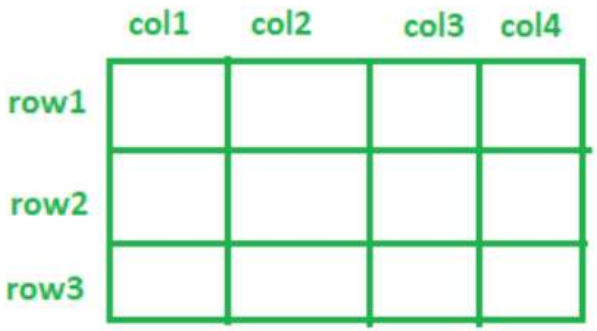
Size of multidimensional arrays: The total number of elements that can be stored in a multidimensional array can be calculated by multiplying the size of all the dimensions.

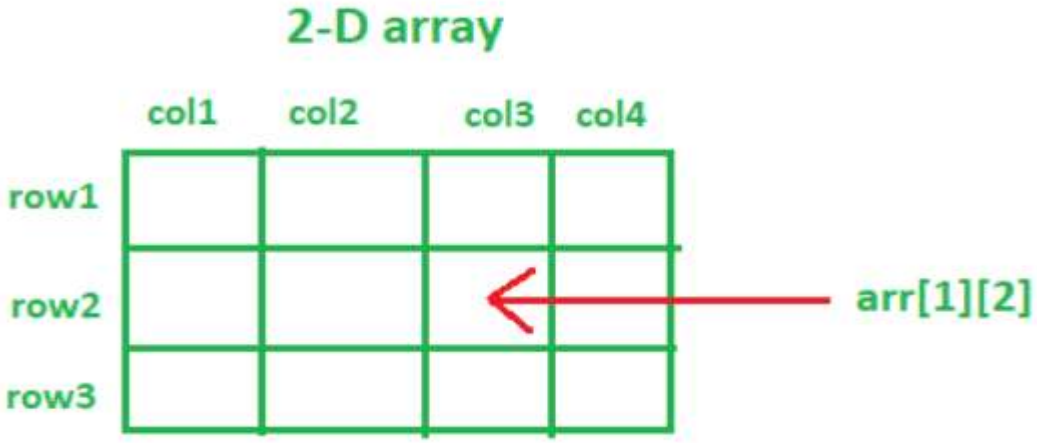
For example: The array **int[][] x = new int[10][20]** can store a total of (10*20) = 200 elements. Similarly, array **int[][][] x = new int[5][10][20]** can store a total of (5*10*20) = 1000 elements.

Representation of 2D array in Tabular Format:

A two – dimensional array can be seen as a table with ‘x’ rows and ‘y’ columns where the row number ranges from 0 to (x-1) and column number ranges from 0 to (y-1). A two – dimensional array ‘x’ with 3 rows and 3 columns is shown below:

	Column 0	Column 1	Column 2
Row 0	x[0][0]	x[0][1]	x[0][2]
Row 1	x[1][0]	x[1][1]	x[1][2]
Row 2	x[2][0]	x[2][1]	x[2][2]


2-D array
col1
col2
col3
col4
row1
row2
row3


2-D array
col1
col2
col3
col4
row1
row2
row3
arr[1][2]

Declaring 2-D array in Java:

Any 2-dimensional array can be declared as follows:

Syntax:

data_type[][]array_Name=

new *data_type[no_of_rows][no_of_columns];*

```
int[][] twoD_arr = new int[10][20];
```

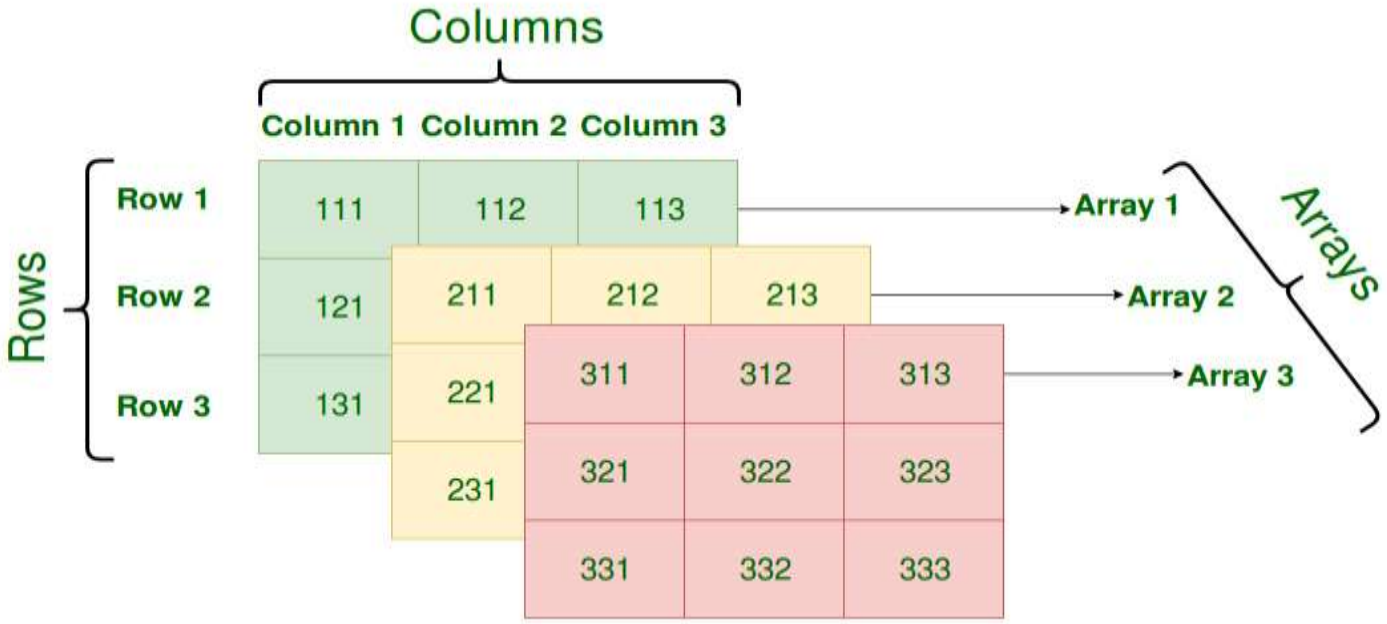
- The total elements in any 2D array will be equal to (no_of_rows) * (no_of_columns).
- **no_of_rows:** The number of rows in an array. e.g. no_of_rows = 3, then the array will have three rows.
- **no_of_columns:** The number of columns in an array. e.g. no_of_columns = 4, then the array will have four columns.

```
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
System.out.println(myNumbers[1][2]);
```

Outputs: 7

Representation of 3D array in Tabular Format:

A three-dimensional array can be seen as a table of arrays with ‘x’ rows and ‘y’ columns where the row number ranges from 0 to (x-1) and column number ranges from 0 to (y-1). A three – dimensional array with 3 array containing 3 rows and 3 columns is shown below:



Declaration – Syntax:

data_type[][][] array_name = new data_type[x][y][z];
For example: int[][][] arr = new int[10][20][30];

Three dimensional array:
int[][][] threeD_arr = new int[10][20][30];

Initialization – Syntax:

array_name[array_index][row_index][column_index] = value;
For example: arr[0][0][0] = 1;

Reference:

- https://www.w3schools.com/java
- https://www.geeksforgeeks.org/java
- https://www.javatpoint.com/
- https://www.programiz.com/

Thank You