**Java Arrays**

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

String[] cars;

String[] cars= new String[5]; //String array declaration with size

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

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"}; This is hard-coded

We can edit array elements.

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

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

how to access

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

System.out.println(cars[0]);

// Outputs Volvo

We can assign other value to array elements.

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

cars[0] = "Opel";

System.out.println(cars[0]);

// Now outputs Opel instead of Volvo

Array Methods

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

System.out.println(cars.length);

// Outputs 4

The **Arrays** class in **java.util package** is a part of the **Java Collection Framework**. This class provides static methods to dynamically create and access **Java arrays**. It consists of only static methods and the methods of Object class. The methods of this class can be used by the class name itself.

The class hierarchy is as follows:

java.lang.Object

? java.util.Arrays

**Array Sort**

int[] array = {3, 2, 1};

Arrays.sort(array);

for (int i : array) {

System.out.print(i + " ");

}

### **Arrays equals**

int[] array1 = {1, 2, 3};

int[] array2 = {1, 2, 3};

boolean isEqual = Arrays.equals(array1, array2);

System.out.println(isEqual);

### **Arrays binarySearch**

but requires the array to be sorted first.

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

int index = Arrays.binarySearch(array, 3);

System.out.println(index);

## Exploring Alternative Approaches to Array Manipulation

### Using ArrayLists for Dynamic Arrays

While Java arrays are powerful, they have a limitation: their size is fixed at the time of creation. To overcome this, we can use ArrayList, a resizable array implementation in the Java Collections Framework.

# Java Multidimensional Arrays

## How to initialize a 2d array in Java?

Here is how we can initialize a 2-dimensional array in Java.

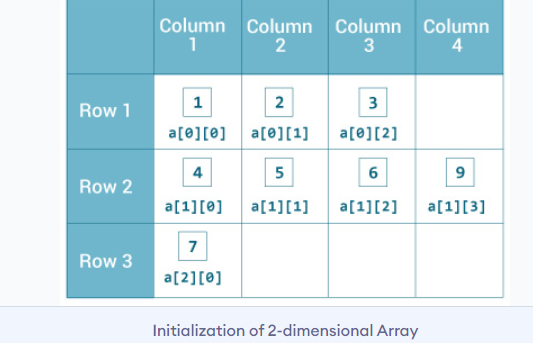
int[][] a = {

{1, 2, 3},

{4, 5, 6, 9},

{7},

};



### **Example: Print all elements of 2d array Using Loop**

class MultidimensionalArray {

public static void main(String[] args) {

int[][] a = {

{1, -2, 3},

{-4, -5, 6, 9},

{7},

};

for (int i = 0; i < a.length; ++i) {

for(int j = 0; j < a[i].length; ++j) {

System.out.println(a[i][j]);

}

}

}

}

[Run Code](https://www.programiz.com/java-programming/online-compiler" \t "https://www.programiz.com/java-programming/_blank)

****Output****:

1

-2

3

-4

-5

6

9

7

Mostly you will see questions related to Multidimentional Arrays.

Lets Recall

1. Declare define and intilize an array and use all methods.
2. Arrays class and when to use and its methods
3. 2D-array print all elements and Practice more 2D array questions.