chandru1532002@gmail.com

Arrays & Arrays Class in Java

An **Array** is a fixed-size data structure that stores multiple values of the same type in contiguous memory locations.

The **Arrays class** (from java.util.Arrays) provides utility methods for sorting, searching, comparing, and manipulating arrays.

01

Definition!

Declaring & Initializing an Array

```
public class ArrayExample {
    public static void main(String[] args) {
        int[] numbers = {10, 20, 30, 40, 50};

        System.out.println("First element: " +
        numbers[0]);

        System.out.println("Array elements:");
        for (int num : numbers) {
            System.out.print(num + " ");
        }
    }
}
```

Arrays in Java

Multidimensional Arrays

```
public class MultiDimArray {
    public static void main(String[] args) {
        int[][] matrix = { {1, 2, 3}, {4, 5, 6} };

        System.out.println("Element at [1][2]: "
+ matrix[1][2]); // Output: 6
    }
}
```

The **Arrays class** provides useful static methods to operate on arrays.

Common Methods of Arrays Class:

✓ **sort()** – Sorts an array

✓ **binarySearch()** – Searches for an element (sorted array)

✓ **copyOf()** – Copies elements to a new array

✓ **fill()** – Fills the entire array with a value

✓ **equals()** – Compares two arrays

✓ **toString()** – Converts array to string

04

Arrays Class in Java

Sorting an Array using Arrays.sort()

```
public class SortArray {
    public static void main(String[] args) {
        int[] numbers = {5, 2, 8, 1, 3};
        Arrays.sort(numbers);

                    System.out.println("Sorted Array: " +
Arrays.toString(numbers));
    }
}
```

05

Searching an Element

```
public class SearchArray {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5, 6};

        int index = Arrays.binarySearch(numbers, 4);
        System.out.println("Index of 4: " + index);
    }
}
```

Copying an Array using Arrays.copyOf()

```
public class CopyArray {
    public static void main(String[] args) {
        int[] original = {1, 2, 3, 4, 5};
        int[] copied = Arrays.copyOf(original, original.length);

        System.out.println("Copied Array: " +
        Arrays.toString(copied));
    }
}
```

07

Filling an Array using Arrays.fill()

```
import java.util.Arrays;

public class FillArray {
    public static void main(String[] args) {
        int[] numbers = new int[5];
        Arrays.fill(numbers, 10);

                          System.out.println("Filled   Array:   "   +
Arrays.toString(numbers));
    }
}
```

Comparing Two Arrays using Arrays.equals()

```
public class CompareArrays {
    public static void main(String[] args) {
        int[] array1 = {1, 2, 3};
        int[] array2 = {1, 2, 3};
        int[] array3 = {3, 2, 1};

              System.out.println("Array1 equals Array2: " +
Arrays.equals(array1, array2));
              System.out.println("Array1 equals Array3: " +
Arrays.equals(array1, array3));
    }
}
```