

Java Arrays - Detailed Notes

1. Introduction to Arrays in Java

An array in Java is a data structure that holds a fixed number of values of a single type.

The length of an array is established when the array is created. After creation, its length is fixed.

Syntax:

```
dataType[] arrayName = new dataType[size];
```

```
int[] numbers = new int[5];  
numbers[0] = 10;  
numbers[1] = 20;  
numbers[2] = 30;  
numbers[3] = 40;  
numbers[4] = 50;  
System.out.println(numbers[2]);
```

Output: 30

2. Array Initialization and Access

Arrays can also be initialized with values directly:

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

You can access elements using their index (starting from 0).

```
int[] nums = {1, 2, 3, 4, 5};  
System.out.println(nums[0]);
```

Output: 1

Java Arrays - Detailed Notes

3. Looping through Arrays

You can use loops to traverse arrays, commonly using for or enhanced for loops.

```
int[] arr = {10, 20, 30};  
for (int i = 0; i < arr.length; i++) {  
    System.out.println(arr[i]);  
}
```

Output: 10 20 30

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

Output: 10 20 30

4. Advantages of Arrays

- Arrays allow random access of elements using indices.
- Arrays are easy to traverse and manipulate using loops.
- Arrays help in efficient memory allocation when the size is known.
- They store multiple values in a single variable, reducing code complexity.
- Arrays are faster in accessing and modifying data compared to some data structures.

5. Two-Dimensional Arrays

2D arrays are arrays of arrays. They are useful for matrix-like data representation.

Syntax:

Java Arrays - Detailed Notes

```
dataType[][] arrayName = new dataType[rows][columns];
```

```
int[][] matrix = {  
    {1, 2, 3},  
    {4, 5, 6},  
    {7, 8, 9}  
};  
  
System.out.println(matrix[1][1]);
```

Output: 5

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

Output: 1 2 3 4 5 6 7 8 9