Java Arrays - Methods, Examples & Real-Time Use Cases

Java Arrays Methods with Examples

1. Arrays.toString()

Converts the array to a string format.

```
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4};
        System.out.println(Arrays.toString(arr)); // [1, 2, 3, 4]
    }
}
```

2. Arrays.sort()

Sorts the array in ascending order.

```
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        int[] arr = {5, 1, 4, 2};
        Arrays.sort(arr);
        System.out.println(Arrays.toString(arr)); // [1, 2, 4, 5]
    }
}
```

3. Arrays.copyOf()

Copies the specified array, truncating or padding with default values.

```
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        int[] original = {1, 2, 3};
        int[] copy = Arrays.copyOf(original, 5);
        System.out.println(Arrays.toString(copy)); // [1, 2, 3, 0, 0]
    }
}
```

4. Arrays.equals()

Checks if two arrays are equal.

```
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        int[] a = {1, 2, 3};
        int[] b = {1, 2, 3};
        System.out.println(Arrays.equals(a, b)); // true
    }
}
```

5. Arrays.fill()

Fills the array with a specific value.

```
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        int[] arr = new int[5];
        Arrays.fill(arr, 7);
        System.out.println(Arrays.toString(arr)); // [7, 7, 7, 7, 7]
    }
}
```

Real-Time Use Cases of Arrays

Use Case 1: Finding the missing number in a range

```
import java.util.*;

public class MissingNumber {
    public static void main(String[] args) {
        int[] nums = {1, 2, 4, 6, 3, 7, 8};
        int n = 8;
        int sum = (n * (n + 1)) / 2;
        for (int num : nums) {
            sum -= num;
        }
        System.out.println("Missing number: " + sum); // Output: 5
    }
}
```

Use Case 2: Remove duplicates from sorted array

```
import java.util.Arrays;

public class RemoveDuplicates {
    public static int removeDuplicates(int[] nums) {
        int i = 0;
        for (int j = 1; j < nums.length; j++) {
            if (nums[i] != nums[j]) {
                i++;
                nums[i] = nums[j];
            }
        }
        return i + 1;
    }
    public static void main(String[] args) {
        int[] arr = {1, 1, 2, 2, 3};
        int len = removeDuplicates(arr);
        for (int i = 0; i < len; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

Use Case 3: Two Sum Problem

```
import java.util.*;

public class TwoSum {
    public static void main(String[] args) {
        int[] nums = {2, 7, 11, 15};
        int target = 9;
        Map<Integer, Integer> map = new HashMap<>();
        for (int i = 0; i < nums.length; i++) {
            int complement = target - nums[i];
            if (map.containsKey(complement)) {
                System.out.println("Indices: " + map.get(complement) + ", " + i);
                break;
            }
            map.put(nums[i], i);
        }
    }
}
```