util.Arrays vs reflect.Array in Java with Examples

Last Updated : 21 Jan, 2022

Array class in **java.lang.reflect package** is a part of the **Java Reflection**. This class provides static methods to create and access Java arrays dynamically. It is a final class, which means it can't be instantiated or changed. Only the methods of this class can be used by the class name itself. On the other hand, 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.

Let us directly discuss major differences via table as follows on basic of few factor as listed:

Difference between Array and Arrays

Basic	Array	Arrays
Package Existence in class hierrarchy	The Array class exists in the java.lang.reflect package	The Arrays class exists in java.util package

Class Hierrarchy	`java.lang.Object`
	`↳ java.lang.reflect`
	`↳ Class Array`
		`java.lang.Object`
		`↳ java.util`
		`↳ Class Arrays`

Immutability	The Array class is immutable in nature	Arrays class is not immutable in nature. By immutable, it means that the class cannot be extended or inherited. The Array class is declared as final to achieve immutability.
Class declaration	`public final class Array extends Object`	`public class Arrays extends Object`

Basic	Array	Arrays
Usage	**Array class** provides static methods to dynamically create and access Java arrays. This Array class keeps the array to be type-safe.	**Arrays class** contains various methods for manipulating arrays (such as sorting and searching)

Implementation:

```
// Java program to Illustrate Usage of Array class
// vs Arrays Class

// Importing both classes from resprective packages
import java.lang.reflect.Array;
import java.util.Arrays;

// Main class
public class GFG {

    // Main driver method
    public static void main(String[] args)
    {

        // Getting the size of the array
        int[] intArray = new int[5];

        // Adding elements into the array
        // using setInt() method of Array class
        Array.setInt(intArray, 0, 10);

        // Printing the Array content
        // using util.Arrays class
        System.out.println(Arrays.toString(intArray));
    }
}
```

Output:

```
[10, 0, 0, 0, 0]
```

Only Java Can Get
The Job Done For You.

So Strengthen Your Foundations and

Start Learning