GUVI | HCL
Skill Up. Level Up

Java Day 3

GUVI | HCL

Skill Up. Level Up

Arrays in Java

Arrays

Arrays are fundamental structures in Java that allow us to store multiple values of the same type in a single variable.

They are useful for storing and managing collections of data.

Arrays in Java are objects, which makes them work differently from arrays in C/C++ in terms of memory management.

Example

```
public class Main {    public static void main(String[] args)

    {

// initializing array

        int[] arr = { 1, 2, 3, 4, 5 }.

        int n = arr.length   // size of array

        for (int i = 0; i < n; i++)   // traversing array

            System.out.print(arr[i] + " ");

    }

}
```

Basics of Arrays in Java

1. Array Declaration

2.Create an array

3. Access an Element of an Array

4. Change an Array Element

5. Array Length

1. Array Declaration

To declare an array in Java, use the following syntax:

type[] arrayName;

type: The data type of the array elements (e.g., int, String).

arrayName: The name of the array.

Note: The array is not yet initialized.

2. Create an Array

To create an array, you need to allocate memory for it using the new keyword:

```
// Creating an array of 5 integers

int[] numbers = new int[5];
```

This statement initializes the numbers array to hold 5 integers. The default value for each element is 0.

3. Access an Element of an Array

We can access array elements using their index, which starts from 0:

```
// Setting the first element of the array

numbers[0] = 10;

// Accessing the first element

int firstElement = numbers[0];
```

The first line sets the value of the first element to 10. The second line retrieves the value of the first element.

4. Change an Array Element

To change an element, assign a new value to a specific index:

```
// Changing the first element to 20

numbers[0] = 20;
```

5. Array Length

We can get the length of an array using the length property:

```
// Getting the length of the array

int length = numbers.length;
```

Now, we have completed with basic operations so let us go through the in-depth concepts of Java Arrays, through the diagrams, examples, and explanations.

Example:

```
class Main {

    public static void main(String[] args)

    {

        // declares an Array of integers.

        int[] arr;



        // allocating memory for 5 integers.

        arr = new int[5];
```

```
// initialize the elements of the array

        // first to last(fifth) element

          arr[0] = 10;

        arr[1] = 20;

        arr[2] = 30;

        arr[3] = 40;

        arr[4] = 50;
```

```
// accessing the elements of the specified array
        for (int i = 0; i < arr.length; i++)
            System.out.println("Element at index "
                               + i + " : " + arr[i]);
    }
}
```

Output

Element at index 0 : 10

Element at index 1 : 20

Element at index 2 : 30

Element at index 3 : 40

Element at index 4 : 50

GUVI | HCL
Skill Up. Level Up

Types of Arrays in Java

1. Single-Dimensional Arrays

These are the most common type of arrays, where elements are stored in a linear order.

```
// A single-dimensional array

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

2. Multi-Dimensional Arrays

Arrays with more than one dimension, such as two-dimensional arrays (matrices).

```
// A 2D array (matrix)
int[][] multiDimArray = {
     {1, 2, 3},
    {4, 5, 6},
    {7, 8, 9} };
```

Arrays of Objects in Java

An array of objects is created like an array of primitive-type data items in the following way.

Syntax:

Method 1:

ObjectType[] arrName;

Method 2:

ObjectType arrName[];

Example of array

```
class Student {

    public int roll_no;

    public String name;

    Student(int roll_no, String name){

        this.roll_no = roll_no;

        this.name = name;

    }

}
```

```
public class Main {

    public static void main(String[] args){


        // declares an Array of Student

        Student[] arr;


        // allocating memory for 5 objects of type Student.

        arr = new Student[5];
```

```
// initialize the elements of the array
        arr[0] = new Student(1, "aman");
        arr[1] = new Student(2, "vaibhav");
        arr[2] = new Student(3, "shikar");
        arr[3] = new Student(4, "dharmesh");
        arr[4] = new Student(5, "mohit");
```

```
// accessing the elements of the specified array

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

            System.out.println("Element at " + i + " : { "

                                + arr[i].roll_no + " "

                                + arr[i].name+" }");

    }

}
```

Output

Element at 0 : { 1 aman }

Element at 1 : { 2 vaibhav }

Element at 2 : { 3 shikar }

Element at 3 : { 4 dharmesh }

Element at 4 : { 5 mohit }

What happens if we try to access elements outside the array size?

JVM throws ArrayIndexOutOfBoundsException to indicate that the array has been accessed with an illegal index. The index is either negative or greater than or equal to the size of an array.

```
public class Main{

    public static void main(String[] args)

    {

        int[] arr = new int[4];

        arr[0] = 10;

        arr[1] = 20;

        arr[2] = 30;

        arr[3] = 40;
```

```
System.out.println(
            "Trying to access element outside the size of array");
        System.out.println(arr[5]);
    }
}
```

Output

Trying to access element outside the size of array

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 4

at GFG.main(GFG.java:13)