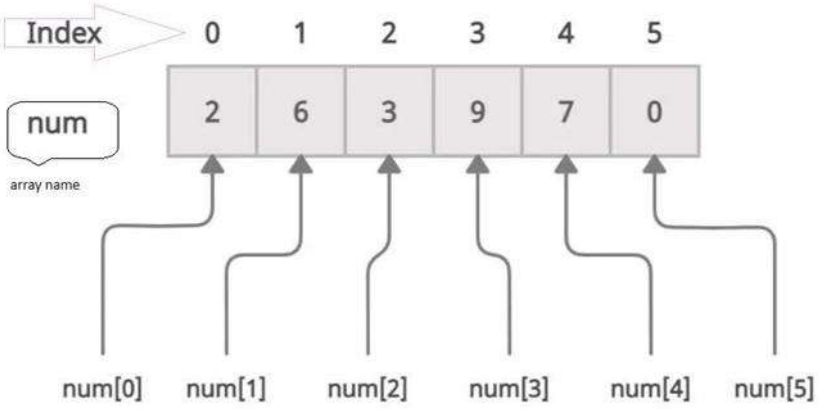
Java Arrays

Arrays are one of the most fundamental and widely used data structures in computer programming. They allow for the storage of multiple values, typically of the same type, in a single data structure. In Java, arrays are objects, which provide methods to interact with the data they store.



Introduction to Java Arrays:

An array is a fixed-size, ordered collection of elements that can store homogeneous type of data. Each element in an array has an associated index, starting from 0, that allows for easy access. The main characteristic that distinguishes arrays from other data structures is their static nature once initialized, the size of an array cannot be changed.

Constructing Arrays: Syntax and Initialization

Arrays are declared by specifying the data type followed by square brackets and can be instantiated using **new** keyword.

```
int[] numbers = new int[5]; // Initialization with size
```

Alternatively, declaration, construction, and initialization can be done:

```
int [] numbers ={10,11,12,13,14} // Initialization with elements
```

Accessing Array Elements:

Each element in the array has unique index, starting from 0 and going up to array.length

Looping Through Arrays:

Using a traditional for loop:

```
int[] number= {11, 12, 13, 14,15};
for (int i = 0; i < number.length; i++) {
   System.out.println(number[i]);
}
```

Using an enhanced for loop (for-each loop):

```
for (int i: number) {
System.out.println(i);
```

```
}
```

Multidimensional Arrays:

Java supports multidimensional arrays, which are essentially arrays of arrays. The most common example is a two-dimensional array or a matrix.

Creating a 2D Array**:**

```
int [][] arr=new int[2][3]; // Initialization with size
Int [][] ages={{20,21,22},{23,24,25}}; // Initialization with values
```

Looping through a 2D array involves nested loops:

```
int [][]ages= {{21,22,24},{23,24,25}};
```

Using a traditional for loop:

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

Using an enhanced for loop (for-each loop):

```
for(int[]i:ages){
        for(int value:i) {
             System.out.println(value);
   }
}
```

Common Operations on Arrays:

Java provides java.util.Arrays class which provides various utility methods for arrays.

Sorting: Arrays.sort(); Sorts the specified array into ascending numerical order.

Searching: Arrays.binarySearch(); returns the index of the element if found, else it returns a negative value.

Copying: Arrays.copyOf(,); returns a copy of the original array, truncated or padded with zeros to obtain the specified length

Disadvantages of Array:

1. Arrays can store only homogeneous type of data.
2. Arrays can not grow or shrink in size. i.e. fixed in size
3. Arrays require contiguous memory allocation.