VIPS
Technical Campus
योग: कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

SCHOOL OF ENGINEERING AND TECHNOLOGY

Course : OBJECT ORIENTED PROGRAMMING

Paper Code: AIML-202,IOT-202

Faculty : Dr. Shivanka

Assistant Professor

VIPS

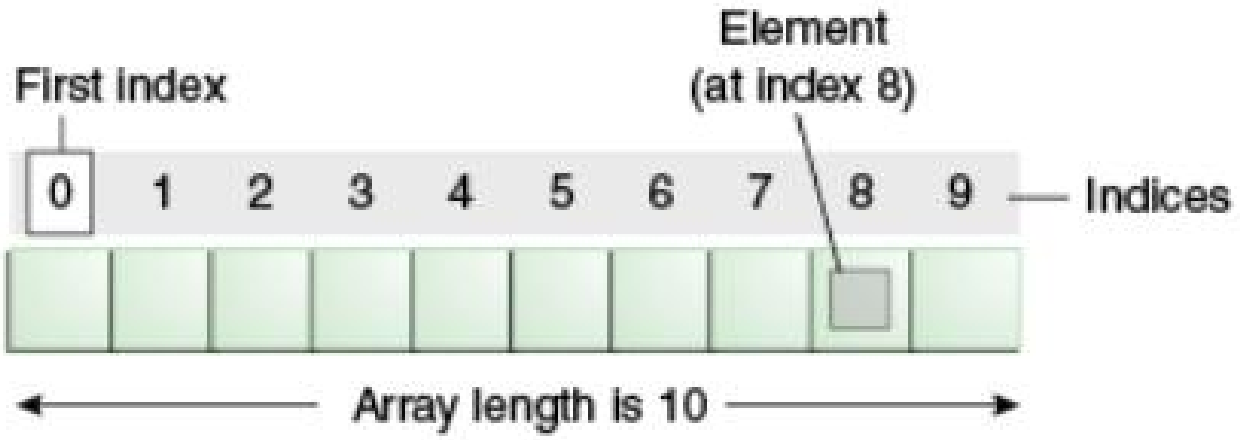
Java Arrays

- Java array is an object which contains elements of a similar data type.
- Additionally, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.
- In Java, array is an object of a dynamically generated class. Java array inherits the Object class, and implements the Serializable as well as Cloneable interfaces. We can store primitive values or objects in an array in Java.

Java Arrays

- Similar to C/C++, we can also create single dimentional or multidimentional arrays in Java.

- Moreover, Java provides the feature of anonymous arrays which is not available in C/C++



Types of Array in java

There are two types of array.

- Single Dimensional Array
- Multidimensional Array

Single Dimensional Array in Java

Syntax to Declare an Array in Java

dataType[] arr; (or)

dataType []arr; (or)

dataType arr[];

Instantiation of an Array in Java

arrayRefVar=new datatype[size];

Example of Java Array

The simple example of java array, where to declare, instantiate, initialize and traverse an array.

/

```
/Java Program to illustrate how to declare, instantiate, initialize  //and traverse the Java array.
class Testarray{
public static void main(String args[]){
int a[]=new int[5];//declaration and instantiation
a[0]=10;//initialization
a[1]=20;
a[2]=70;
a[3]=40;
a[4]=50;
//traversing array
for(int i=0;i<a.length;i++)//length is the property of array
System.out.println(a[i]);
}}
```

output

```
10
20
70
40
70
```

Declaration, Instantiation and Initialization of Java Array

- We can declare, instantiate and initialize the java array together by:

- int a[]={33,3,4,5};
- //declaration, instantiation and initialization

- Java Program to illustrate the use of declaration, instantiation
- //and initialization of Java array in a single line
- class Testarray1{
- public static void main(String args[]){
- int a[]={33,3,4,5};//declaration, instantiation and initialization
- //printing array
- for(int i=0;i<a.length;i++)//length is the property of array
- System.out.println(a[i]);
- }}

Multidimensional Array in Java

- To create a two-dimensional array, add each array within its own set of curly braces:
- dataType[][] arrayRefVar; (or)
- dataType [][]arrayRefVar; (or)
- dataType arrayRefVar[][]; (or)
- dataType []arrayRefVar[];

```
int[][] arr=new int[3][3];
//3 row and 3 column
```

- Example to initialize Multidimensional Array in Java
- arr[0][0]=1;
- arr[0][1]=2;
- arr[0][2]=3;
- arr[1][0]=4;
- arr[1][1]=5;
- arr[1][2]=6;
- arr[2][0]=7;
- arr[2][1]=8;
- arr[2][2]=9;

Multidimensional Array in Java


```
//Java Program to illustrate the use of multidimensional array
class Testarray3{
public static void main(String args[]){
//declaring and initializing 2D array
int arr[][]={{1,2,3},{2,4,5},{4,4,5}};
//To create a two-dimensional array, add each array within its own set of curly braces:
//printing 2D array
for(int i=0;i<3;i++){
 for(int j=0;j<3;j++){
  System.out.print(arr[i][j]+" ");
 }
 System.out.println();
}
}}
```

- Output

```
1 2 3
2 4 5
4 4 5
```

Exp3:-Sum of array of N no element enter by user


```
package array;
import java.util.Scanner;
public class Arraysum {
   public static void main(String[] args)
  { int n,sum=0;   Scanner ob=new
Scanner(System.in);
    System.out.println("Enter the size of array : ");
    n=ob.nextInt();nt a[]=new int[n];
    System.out.println("Enter array Elements : ");
    for(int i=0;i<n;i++)
{
    a[i]=ob.nextInt();
    sum=sum+a[i];
    }
    System.out.println("Sum of array is : "+sum);
    }
}
```

Output: -

```
Enter the size of array :
10
Enter array Elements :
2
4
6
8
10
12
14
16
18
20
Sum of array N element is : 110
```

Java Arrays – The Basics

Declaring an array

```
int[] myArray;
int[] myArray = new int[5];
String[] stringArray = new String[10];
String[] strings = new String[] {“one”, “two”};
```

Checking an arrays length

```
int arrayLength = myArray.length;
```

Looping over an array

```
for(int I=0; I<myArray.length; i++)
{
   String s = myArray[i];
}
```

Java Arrays – Bounds Checking

Bounds checking

- Java does this automatically.
- It is impossible to go beyond the end of an array (unlike C/C++)
- Automatically generates an `ArrayIndexOutOfBoundsException`

```
public class TestArray {
public static void main(String[] args)
 {
 double[] myList = {1.9, 2.9, 3.4, 3.5};
 // Print all the array elements
for (int i = 0; i < myList.length; i++)
{System.out.println(myList[i] + " "); }
// Summing all elements

                double total = 0;
for (int i = 0; i < myList.length; i++)
{       total += myList[i];                     }
System.out.println("Total is " + total);
// Finding the largest element
                double max = myList[0];
                for (int i = 1; i < myList.length;
i++)
                {
        if (myList[i] > max)
max = myList[i];
                }
                System.out.println("Max is " +
max);
}
 }
```

Java Arrays – Copying

- ✓ Don't copy arrays "by hand" by looping over the array
- ✓ The System class has an `arrayCopy` method to do this efficiently

```
int array1[] = new int[10];
int array2[] = new int[10];
```

The System class has an arraycopy method that you can use to efficiently copy data from one array into another:

public static void arraycopy(Object src, int srcPos, Object dest, int destPos, int length)

The two Object arguments specify the array to (array1) copy from and the array to copy to (array2). The three int arguments specify the starting position in the source array, the starting position in the destination array, and the number of array elements to copy

```
//assume we add items to array1

//copy array1 into array2
System.arrayCopy(array1, 0, array2, 0, 10);

//copy last 2 elements to 7 elament in array1 into first 7 of array2
System.arrayCopy(array1,2 , array2, 0, 7);
```

Array copy method

The following program, ArrayCopyDemo, declares an array of String elements. It uses the System.arraycopy method to copy a subsequence of array components into a second array:

```
class ArrayCopyDemo {

    public static void main(String[] args) {

        String[] array1 = {

            "Vrinda", "Amit", "Chahat", "Simran", "Rohit ",   "Deepak", "Angel", " Shagun", "Sheetal", "Srishti", "Mohit",  "Rohit", "Aman" };

          String[] array2 = new String[7];

        System.arraycopy(array1, 2, array2, 0, 7);

        for (String name : array2) {

            System.out.print(name + " ");

        }}}
```

The output from this program is:

Chahat simran Rohit Deepak Angel Shagun Sheetal

Java Arrays Class

The **java.util.Arrays** class contains a static factory that allows arrays to be viewed as lists. Following are the important points about Arrays –

- This class contains various methods for manipulating arrays (such as sorting and searching).
- The methods in this class throw a NullPointerException if the specified array reference is null.

static int binarySearch(byte[] a, byte key) This method searches the specified array of bytes for the specified value using the binary search algorithm.

static boolean equals(long[] a, long[] a2) This method returns true if the two specified arrays of longs are equal to one another.

static int hashCode(byte[] a) This method returns a hash code based on the contents of the specified array.

static void sort(char[] a) This method sorts the specified array of chars into ascending numerical order.

Around **105** methods defined in Arrays class to do different operations on arrays.

Thank You