Array

Array

- Arrays are a collection of data items that have the same data type and referred to by a common name(a single variable name).]
- For example, an Integer array will have all integer elements ,Float array will have float elements and a String array will have Strings for each element of the array.

Some points about arrays:

- Arrays are considered as objects in Java.
- All arrays in Java are dynamically allocated.
- Array organized the data in the form of elements where each element of array store a single data (no. of data = no. of elements in array).
- Each element of array store similar type of data. And the type of data is decided by us during declaration of array.

- To access or identify any element ,array provide unique **index number** for each element i.e started from 0(the first element) till n-1 where n is the length of the array. For example, an array of length 5 would have indices ranging from 0 to 4 (5-1).
- Each element has common name i.e name of the array
- Once we create an array every array element by default initialize with default value .
- Like other variables in Java, an array must be defined before it can be used
- length of array find using the length variable i.e it is not a function.

Declaration of Array

- **Syntax** for declaring an array variable:- To declare an array, define the variable type with **square brackets[]**

  dataType var_name[size];

  or , datatype[size/no. of element] var_name;

  or, datatype [size]var_name ;

- **Example-**

  int [] x;

  int x[];

  int []x

- The size of an array must be an integer value(can not be in decimal).
- Example- int [5] marks; // declare an array

  marks = new int[5] // allocate memory

Here, the array can store **5** elements.(**size or length** of the array is 5).

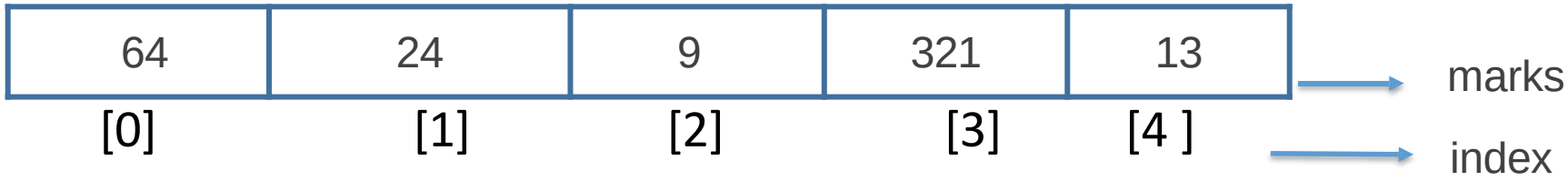
- we can declare and allocate the memory of an array in a single line(using the **new** operator).

Syntax- datatype[] arrayname = new datatype[length/size];

or, datatype arrayname[] = new datatype[lenght];

For example - int[] marks = new int[5];

Array length=5
Start index=0
Last index=4



- we can randomly access the elements of an array using their index number.

  marks[0]

  marks[1]

  marks[2]

  marks[3]

  marks[4]

- The size of the array cannot be changed (once initialized).
- Array elements can be accessed by a for loop and a for each loop. But elements in the array cannot be altered by using the for each loop.

- **Initialize Arrays -** by two ways

a. Syntax- datatype [] arrayname={list of value with , separator };

Example- int[] myArray= {1,2,3,4,5} ; // declare and initialize

b. int[] myArray = new int[5];

myArray[0] = 1 ; // initialize array

myArray[1] = 2 ;

myArray[2] = 3 ;

myArray[3] = 4 ;

myArray[4] = 5 ;

How to Access Elements of an Array

- access the element of an array using their index number

```
public class Array1 {

    public static void main(String[] args) {
        int[] arr = new int[3]; // declare and create 1D array
        arr[0]=34; // initialize first element
        arr[1]=39; // initialize second element
        arr[2]=47; // initialize third element

        // each element of array is accessed by its unique index value
        System.out.println("Accessing Elements of array: ");
        System.out.println("Element 1 at index 0:"+ arr[0]);
        System.out.println("Element 2 at index 1:"+ arr[1]);
        System.out.println("Element 3 at index 2:"+ arr[2]);

        arr[1]=23;// change the array element
        System.out.println("Element 2 at index 1 after change:"+ arr[1]);
    }

}
```

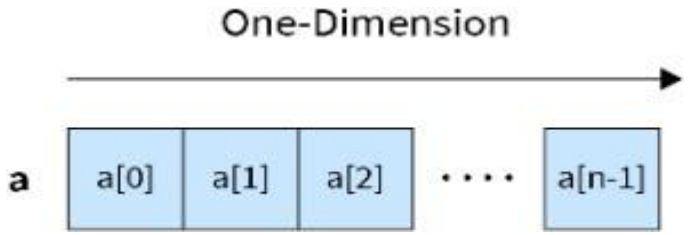
- We can also access all the elements of array at once by using loop (for loop and for each loop).For example-

```
public class Array1 {

    public static void main(String[] args) {
        // declare, create and initialize in a single line
        String strArray[] = {"Python","java","C","C++","PHP"};

        //find the size of the array
        int len= strArray.length;
        System.out.println("Size of array: "+len);
        // access using for loop
        for(int i=0;i<len;i++)
            System.out.println(strArray[i]);

        // access using for each loop
        for(String a:strArray)
            System.out.println(a);


    }

}
```

Types of Array

- Single-Dimensional Array
- Two-dimensional Array
- Multi-Dimensional Array

1. Single Dimensional array- In this , data are arranged or store in one direction either horizontally or vertically.



In two ways we can create an array:-

➢ Array Creation with values-

Syntax- datatype [] arrayname={list of value with , separator };

Example- int[] arr = {10, 20, 30, 40};

```java
public class Array1 {

    public static void main(String[] args) {
        // declare, create and initialize in a single line
        double arr[]= {12,98,78.9,45,62.97,24};

        //find the size of the array
        int len= arr.length;
        System.out.println("Size of array: "+len);
        // access using for loop
        for(int i=0;i<len;i++)
            System.out.println(arr[i]);

        // access using for each loop
        for(double a:arr)
            System.out.println(a);


    }

}
```


- Array Creation without Values-

Syntax:

```
datatype arrayname[] = new datatype[size];
                Or
datatype arrayname[];
arrayname = new datatype[size];
```

Example - int arr[] = new int[4];

2. Multi-Dimensional Array :– Data is stored in row and column i.e in two direction

In two ways we can create an array:-

- **Array Creation with values-**

Syntax –

datatype[][] arrayname = {{1st row value with , separator },{2nd row value with , separator },{3rd row value with , separator },……};

Example-

```
int[][] a= {{1,2,3,4,5},{6,7,8,9,10}} ;     // declare and initialize the array
```

```java
public class Array1 {

    public static void main(String[] args) {
        // declare, create and initialize in a single line
        int arr[][]= {{1,2,3,4,5},{6,7,8,9,10}};

        //find the size of the array
        int len= arr.length;
        System.out.println("Size of array: "+len);
        int len2= arr[0].length;
        System.out.println("Size of array: "+len2);

        // access using for loop
        for(int i=0;i<len;i++) {
            for(int j=0;j<len2;j++) {
                System.out.print(arr[i][j]+" ");
            }
            System.out.println();
        }

        System.out.println("----------------------");
        // access using for each loop
        for(int[] a1:arr) {
            for(int a2:a1) {
                System.out.print(a2+" ");
            }
            System.out.println();
        }

    }
}
```

Array Creation without values

Syntax - datatype[][] arrayname = new datatype[no. of rows][no.of column];

Example- int[][] a = new int [2][5];

int [][]a = new int [2][5];

int a[][] = new int [2][5];

Question - WAP to input 10 data in an array and print the biggest data.

```java
public class Array1 {

    public static void main(String[] args) {
        // declare, create and initialize in a single line
        int arr[][]= new int[2][5];
        //initialize array
        arr[0][0]=5;
        arr[0][1]=15;
        arr[0][2]=12;
        arr[0][3]=95;
        arr[0][4]=65;
        arr[1][0]=4;
        arr[1][1]=91;
        arr[1][2]=07;
        arr[1][3]=39;
        arr[1][4]=32;
        // access using for each loop
        for(int[] a1:arr) {
            for(int a2:a1) {
                System.out.print(a2+" ");
            }
            System.out.println();
        }
    }
}
```

Advantages of Arrays -

- we can access any element randomly from the array with the help of indexes.
- we can store multiple values by using single variable.
- It is easy to store and manipulate large data sets .
- we can retrieve or sort the data efficiently.

Disadvantages of Arrays -

- The size of the array cannot be increased or decreased once it is declared—arrays have a fixed size
- cannot store heterogeneous data.

Anonymous Arrays

- An array created without a name .
- The main purpose of an anonymous array is just for one-time usage.
- An anonymous array is passed as an argument of a method or for temporary operation.
- **Do not have a reference** variable

- **Syntax –**

  methodName(new datatype[] { value1, value2,…. });

- Example –

  new String[] {"Geeks", "for", "Geeks"};

Example of Anonymous Arrays

```
class AnonymousArrayExample {
        // Method that accepts an array
        static void printArray(int[] arr) {
                for (int i : arr) {
                    System.out.print(i + " ");
                }
        }
        public static void main(String[] args) {
             // Anonymous array passed to method
            printArray(new int[] {10, 20, 30, 40});
        }
}
```

Arrays class

- Arrays class is a part of the **Java Collection Framework**.
- java.util package
- provides only static methods (such as sorting and searching).
- **Methods in Arrays Class:-**

Methods	Description
compare()	Compares two arrays [0 – equal , return 1 if not equal]
sort()	Sorts an array
toString()	Converts an array to a string
binarySearch()	Searches for a value in a sorted array

Example

```
import java.util.Arrays;
public class ArraysExample {
        public static void main(String[] args) {
                int[] arr = {5, 2, 9, 1, 3};
                int[] arr2 = {1, 2, 3, 5, 9};
                int[] arr3 = {1, 2, 4, 5, 9};

                // 1. sort() - Sort the array
                Arrays.sort(arr);
                 // 2. equals() - Compare two arrays
                System.out.println("arr equals arr2: " + Arrays.equals(arr, arr2));
                // 3. compare() - Compare two arrays
                int result = Arrays.compare(arr, arr3);
                System.out.println("Comparison result (arr vs arr3): " + result);

        }
}
```

var-args method

- **varargs** (variable-length argument lists) allows a method to accept zero or more arguments of a specified type
- Syntax –

```
returnType methodName(dataType... variableName) {
  // method body
}
```


NOTE –

- Varargs must be the last parameter , if there is other parameter in method
- Only one varargs argument in method

Example of var-args method

```
class VarargsExample {
        static void showNumbers(int... numbers) {
                for (int n : numbers) {
                    System.out.print(n + " ");
                }
        }
        public static void main(String[] args) {
                showNumbers(); // no arguments
                System.out.println();
                showNumbers(1); // one argument
                System.out.println();
                showNumbers(1, 2, 3, 4); // multiple arguments
        }
}
```

Assignment

1. WAP to input 10 data in an array and print sum of the all elements.
2. WAP to input 10 data in an array and search a given particular data from this array.
3. WAP to input some numeric data in an array from the user at the run time and print positions of all those element which is less than 50.
4. WAP to input some numeric data in an array from the user at the run time and find the second biggest data without arranging that array.
5. WAP to input 10 data in an array and arrange the data in ascending order

4. WAP to input 10 data in an array and print the element of array in reverse.

5. Write a program to print even and odd number from the array list.
6. Write a program in Java to merge two arrays into a single array.
7. Write a Java program to find the duplicate values of an array of integer values.
8. Write a program to FIND HOW MANY TIMES ELEMENT IS OCCURED in a given array?
9. WAP to input 10 data in an array and arrange all even data one side.
10. WAP to input data in 5X5 array and print data of that row which sum is even numbers.

THANK YOU!!

C-DAC Patna