

BCA – 401: Java Programming

Rahul Kumar Singh

In today's Class we have discussed on Multidimensional Arrays in Java.

Multidimensional Array in Java:-

In such case, data is stored in row and column based index (also known as matrix form).

Syntax to Declare Multidimensional Array in Java:-

`dataType[][] arrayRefVar; (or)`

`dataType [][]arrayRefVar; (or)`

`dataType arrayRefVar[][]; (or)`

`dataType []arrayRefVar[];`

Example to instantiate Multidimensional Array in Java:-

`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;
```

Example of Multidimensional Java Array:-

Let's see the simple example to declare, instantiate, initialize and print the 2Dimensional array.

//Java Program to illustrate the use of multidimensional array.

```
class Testarray
{
    public static void main(String args[])
    {
        //declaring and initializing 2D array
        int arr[][]={{1,2,3},{2,4,5},{4,4,5}};
        //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
```

Jagged Array in Java:-

If we are creating odd number of columns in a 2D array, it is known as a jagged array. In other words, it is an array of arrays with different number of columns.

Example:-

//Java Program to illustrate the jagged array

```
class TestJaggedArray  
{  
    public static void main(String[] args)  
    {  
        //declaring a 2D array with odd columns  
        int arr[][] = new int[3][];
```

```
arr[0] = new int[3];
arr[1] = new int[4];
arr[2] = new int[2];
//initializing a jagged array
int count = 0;
for (int i=0; i<arr.length; i++)
    for(int j=0; j<arr[i].length; j++)
        arr[i][j] = count++;
//printing the data of a jagged array
for (int i=0; i<arr.length; i++)
{
    for (int j=0; j<arr[i].length; j++)
    {
        System.out.print(arr[i][j]+" ");
    }
    System.out.println();    //new line
}
}
```

Output:-

0 1 2

3 4 5 6

7 8

What is the class name of Java array?

In Java, an array is an object. For array object, a proxy class is created whose name can be obtained by `getClass().getName()` method on the object.

Example:-

//Java Program to get the class name of array in Java

```
class Testarray
```

```
{
```

```
public static void main(String args[])
```

```
{
```

```
//declaration and initialization of array
```

```
int arr[]={4,4,5};
```

```
//getting the class name of Java array
```

```
Class c=arr.getClass();
```

```
String name=c.getName();
```

```
//printing the class name of Java array
```

```
System.out.println(name);  
}  
}
```

Output:-

|

Copying a Java Array:-

We can copy an array to another by the `arraycopy()` method of `System` class.

Syntax of arraycopy method:-

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

Example of Copying an Array in Java:-

//Java Program to copy a source array into a destination array in Java.

```
class TestArrayCopyDemo  
{  
    public static void main(String[] args)  
    {  
        //declaring a source array  
        char[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'f', 'e', 'i', 'n', 'a', 't', 'e',  
'd' };
```

```
//declaring a destination array
char[] copyTo = new char[7];
//copying array using System.arraycopy() method
System.arraycopy(copyFrom, 2, copyTo, 0, 7);
//printing the destination array
System.out.println(String.valueOf(copyTo));
}
}
```

Output:-

caffeine

Cloning an Array in Java:-

Since, Java array implements the Cloneable interface, we can create the clone of the Java array. If we create the clone of a single-dimensional array, it creates the deep copy of the Java array. It means, it will copy the actual value. But, if we create the clone of a multidimensional array, it creates the shallow copy of the Java array which means it copies the references.

Example:-

//Java Program to clone the array

class Testarray

```
{  
public static void main(String args[])  
{  
int arr[]={33,3,4,5};  
System.out.println("Printing original array:");  
for(int i:arr)  
System.out.println(i);  
System.out.println("Printing clone of the array:");  
int carr[]=arr.clone();  
for(int i:carr)  
System.out.println(i);  
System.out.println("Are both equal?");  
System.out.println(arr==carr);  
}  
}
```

Output:-

Printing original array:

33

3

4

5

Printing clone of the array:

33

3

4

5

Are both equal?

false

Addition of 2 Matrices in Java:-

Let's see a simple example that adds two matrices.

//Java Program to demonstrate the addition of two matrices in Java.

```
class Testarray
```

```
{
```

```
public static void main(String args[])
```

```
{
```

```
//creating two matrices
```

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

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

```
//creating another matrix to store the sum of two matrices
int c[][]=new int[2][3];
//adding and printing addition of 2 matrices
for(int i=0;i<2;i++)
{
for(int j=0;j<3;j++)
{
c[i][j]=a[i][j]+b[i][j];
System.out.print(c[i][j]+" ");
}
System.out.println();      //new line
}
}
}
```

Output:-

2 6 8

6 8 10

Multiplication of 2 Matrices in Java:-

In the case of matrix multiplication, a one-row element of the first matrix is multiplied by all the columns of the

second matrix which can be understood by the image given below.

$$\text{Matrix 1} \left\{ \begin{array}{ccc} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{array} \right\} \quad \text{Matrix 2} \left\{ \begin{array}{ccc} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{array} \right\}$$

$$\begin{array}{l} \text{Matrix 1} \\ * \\ \text{Matrix 2} \end{array} \left\{ \begin{array}{ccc} 1*1+1*2+1*3 & 1*1+1*2+1*3 & 1*1+1*2+1*3 \\ 2*1+2*2+2*3 & 2*1+2*2+2*3 & 2*1+2*2+2*3 \\ 3*1+3*2+3*3 & 3*1+3*2+3*3 & 3*1+3*2+3*3 \end{array} \right\}$$

$$\begin{array}{l} \text{Matrix 1} \\ * \\ \text{Matrix 2} \end{array} \left\{ \begin{array}{ccc} 6 & 6 & 6 \\ 12 & 12 & 12 \\ 18 & 18 & 18 \end{array} \right\}$$

Let's see a simple example to multiply two matrices of 3 rows and 3 columns.

//Java Program to multiply two matrices

```
public class MatrixMultiplicationExample
```

```
{
```

```
public static void main(String args[])
```

```
{
```

```
//creating two matrices
```

```
int a[][]={{1,1,1},{2,2,2},{3,3,3}};
```

```
int b[][]={{1,1,1},{2,2,2},{3,3,3}};
```

//creating another matrix to store the multiplication of two matrices

```
int c[][]=new int[3][3]; //3 rows and 3 columns
```

//multiplying and printing multiplication of 2 matrices

```
for(int i=0;i<3;i++)
```

```
{
```

```
for(int j=0;j<3;j++)
```

```
{
```

```
c[i][j]=0;
```

```
for(int k=0;k<3;k++)
```

```
{
```

```
c[i][j]+=a[i][k]*b[k][j];
```

```
}//end of k loop
```

```
System.out.print(c[i][j]+" ");    //printing matrix element
```

```
}//end of j loop
```

```
System.out.println();    //new line
```

```
} } }
```

Output:-

6 6 6

12 12 12

18 18 18