# Understanding Multidimensional Arrays in Java

A multidimensional array in Java is an array that contains more than one row and column, typically used to represent data in tabular form, like matrices. The most common type of multidimensional array is a 2D array.

## Declaring a 2D Array

You can declare a 2D array as follows:  
  
int[][] arrayName;

## Initializing a 2D Array

You can initialize a 2D array in the following ways:

1. 1. With fixed size:  
    int[][] array = new int[3][4]; // 3 rows, 4 columns
2. 2. With values directly:  
    int[][] array = {  
    {1, 2, 3},  
    {4, 5, 6},  
    {7, 8, 9}  
    };
3. 3. Jagged Arrays (rows of different lengths):  
    int[][] jaggedArray = new int[3][];  
    jaggedArray[0] = new int[2]; // First row has 2 columns  
    jaggedArray[1] = new int[3]; // Second row has 3 columns  
    jaggedArray[2] = new int[4]; // Third row has 4 columns

## Accessing Elements in a 2D Array

Access elements using row and column indices:  
  
int value = array[1][2]; // Accesses the element in the second row, third column  
array[0][0] = 10; // Assigns 10 to the first row, first column

## Traversing a 2D Array

1. 1. Using nested for-loops:

for (int i = 0; i < array.length; i++) { // Iterate over rows  
 for (int j = 0; j < array[i].length; j++) { // Iterate over columns  
 System.out.print(array[i][j] + " ");  
 }  
 System.out.println(); // Move to the next row  
 }

1. 2. Using enhanced for-loops:

for (int[] row : array) {  
 for (int value : row) {  
 System.out.print(value + " ");  
 }  
 System.out.println();  
 }

## Common Operations on 2D Arrays

1. 1. Sum of All Elements:

int sum = 0;  
 for (int[] row : array) {  
 for (int value : row) {  
 sum += value;  
 }  
 }  
 System.out.println("Sum: " + sum);

1. 2. Find the Largest Element:

int max = array[0][0];  
 for (int[] row : array) {  
 for (int value : row) {  
 if (value > max) {  
 max = value;  
 }  
 }  
 }  
 System.out.println("Largest Element: " + max);

1. 3. Transpose of a Matrix:

int[][] transpose = new int[columns][rows];  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array[i].length; j++) {  
 transpose[j][i] = array[i][j];  
 }  
 }

## Example Program: Multiplication Table

public class MultiplicationTable {  
 public static void main(String[] args) {  
 int[][] table = new int[10][10];  
  
 // Populate the table  
 for (int i = 0; i < 10; i++) {  
 for (int j = 0; j < 10; j++) {  
 table[i][j] = (i + 1) \* (j + 1);  
 }  
 }  
  
 // Print the table  
 for (int[] row : table) {  
 for (int value : row) {  
 System.out.printf("%4d", value);  
 }  
 System.out.println();  
 }  
 }  
}