**Day 8**

**Multidimensional Arrays in Java**

[Array-Basics in Java](https://www.geeksforgeeks.org/arrays-in-java/) **Multidimensional Arrays** can be defined in simple words as array of arrays. Data in multidimensional arrays are stored in tabular form (in row major order).

**Syntax:**

***data\_type****[1st dimension][2nd dimension][]..[Nth dimension]****array\_name****=****new data\_type****[size1][size2]….[sizeN];*

**where:**

* **data\_type**: Type of data to be stored in the array. For example: int, char, etc.
* **dimension**: The dimension of the array created. For example: 1D, 2D, etc.
* **array\_name**: Name of the array
* **size1, size2, …, sizeN**: Sizes of the dimensions respectively.

**Examples:**

Two dimensional array:

int[][] twoD\_arr = new int[10][20];

Three dimensional array:

int[][][] threeD\_arr = new int[10][20][30];

**Size of multidimensional arrays**: The total number of elements that can be stored in a multidimensional array can be calculated by multiplying the size of all the dimensions.

**For example:** The array **int[][] x = new int[10][20]** can store a total of (10\*20) = 200 elements. Similarly, array **int[][][] x = new int[5][10][20]** can store a total of (5\*10\*20) = 1000 elements.

**Two – dimensional Array (2D-Array)**

Two – dimensional array is the simplest form of a multidimensional array. A two – dimensional array can be seen as an array of one – dimensional array for easier understanding.

**Indirect Method of Declaration:**

* **Declaration – Syntax:**

**data\_type[][] array\_name = new data\_type[x][y];**

For example: int[][] arr = new int[10][20];

* **Initialization – Syntax:**

**array\_name[row\_index][column\_index] = value;**

For example: arr[0][0] = 1;

**InterviewQuestions**

Question 1: What is an array? Can you change the size of the array once created? [answer]

An array is a fundamental data structure to store objects and primitive values. You have different types of array, like 1-dimension, 2-dimension, and N-dimensional array. No, you cannot change the size of the array once created. If you need a dynamic array, consider using the ArrayList class, which can resize itself.

Array concept interview questions and answers in Java

Question 2: Can you store String in an array of Integer in Java? compile-time error or runtime exception? [answer]

This is a tricky question. The answer is both yes and no. You cannot store a string in an array of primitive int, it will result in a compile-time error as shown below, but if you create an array of Object and assign String[] to it and then try to store Integer object on it.

The compiler won't be able to detect that and it will throw ArrayStoreExcpetion at runtime.

int[] primes = new int[10];

primes[0] = "a"; //compile time error

Object[] names = new String[3];

names[0] = new Integer(0); // ArrayStoreException at runtime

Question 3: What is the difference between ArrayIndexOutfOBounds and ArrayStoreException? [answer]

The ArrayIndexOutOfBoundsException comes when your code tries to access an invalid index for a given array, like a negative index or higher index than length - 1. While, ArrayStoreException comes when you have stored an element of a type other than the type of array, as shown in the above example.

Question 4: Can you use Generics with an array? [answer]

No, you cannot use the Generic feature with an array, that's why sometimes List is a better choice over an array in Java, which is also recommended by Joshua Bloch in his class Java book, Effective Java, a must-read for writing good code in Java.

Question 5 : Is it legal to initialize an array int i[] = {1, 2, 3, 4, 5}; [answer]

Yes, it's perfectly legal. You can create and initialize an array in the same line in Java.

Question 6: Difference between b[] and []b in Java? [answer]

You can declare an array in Java by either prefixing or suffixing[] with a variable. There is not much difference between them if you are not creating more than one variable in one line, but if you do then it creates different types of variables, as shown in the following example :

int a[], b; // first is int array, second is just int variable

int[] c, d; // both c and d are integer array

Question 7: What is a two-dimensional array in Java? [answer]

An array of the array in Java. You can declare them like int[][] primes = new int[3][3] which is a matrix of 3x3.

Question 8: Do you have a three-dimensional array in Java? [answer]

Yes, Java supports the N-dimensional array. Actually, a multi-dimensional array in Java is nothing but an array of array, for example, a two-dimensional array is just an array of a one-dimensional array.

Question 9: How to iterate over an array in Java? [answer]

You can either use classical for loop with the index or advanced foreach loop introduced in Java 5 to iterate over an array in Java. If you need the index to select some element or do something else, use for loop otherwise advanced foreach loop is better. It's less error-prone as you don't need to deal with the index.

Question 10: What is the difference between an array and a linked list? [answer]

One key difference between an array and linked list data structure is, Array requires contiguous memory for its element but linked list elements can be scattered in memory, which means it would be difficult to create a big array but your linked list can grow easily.

An array is good for searching elements if you know the index, but adding and removing elements in an array is expensive as compared to the linked list. If you are interested, you can further check out Data Structures and Algorithms: Deep Dive Using Java course on Udemy to learn more about basic data structure and algorithms.

What is the difference between an array and a linked list?

Question 11: How to sort an array in Java? [answer]

You can sort an array in Java by using the Arrays.sort() method. Arrays is a utility class that contains lots of static utility methods to operate on arrays. This method is overloaded and you can optionally provide a Comparator implementation to sort the array in a custom order.

Question 12: How to copy an array in Java? [answer]

You can either manually copy elements of an array by iterating over them, or you can use System.arrayCopy() method to copy elements from one array to another. This is a powerful method that provides fast copy and also allows you to copy the entire or part of the array.

Question 13: How to access elements of an array in Java? [answer]

You can access elements of an array using the index in Java. It starts from 0, so the first element is stored in location zero and the last element has an index length - 1. Trying to access an invalid index in Java, like a negative index or index higher than size will result in ArrayIndexOutOfBoundsException in Java.

Question 14: How to search an array to check if an element exists there? [answer]

You can search an element inside an array by using either a linear search or binary search. Later is faster but you need to sort the array before performing a binary search on it.

The Arrays class from the java.util package provides a binarySearch() method to search an element in an array. Alternatively, you can also convert the array to ArrayList and use its contains() method to find out if an element exists or not.

But, if you want to do it without using an API method, you can also check out this post to implement the binary search algorithm in Java.

Question 15: Can you make an array volatile in Java? [answer]

This is another tricky question in Java. Yes, you can make an array volatile in Java, but you only make the variable that is pointing to array volatile. If an array is changed by replacing individual elements that happen before the guarantee provided by volatile variables will not hold.

Anyway, if you are seriously preparing for a Java interview then you have to do more than just preparing for array-based questions. I suggest you take a look at this Java Interview Guide, which contains 200+ Java questions and answers, good enough to clear any Java interview.

Question 16: Where does an array stored in memory? [answer]

An array is created in the heap space of JVM memory. Since an array is an object in Java, even if you create an array locally inside a method or block, an object is always allocated memory from the heap.

Now that we have seen many arrays concept-based questions from Java interviews, let's move on to the second part of this article, where we'll see some array-based coding problems.

Btw, If you want to prepare more Java-specific questions then you can also take a look at the Java Programming Interview Exposed, which contains lots of questions from Java and related technology like Maven, JUnit, Servlet, JSP, Android, and others.