Multidimensional Arrays

A multidimensional array is basically an array of arrays.

Arrays can have any number of dimensions. The most common are two-dimensional arrays (2D).

Two-Dimensional Arrays

To create a 2D array, add each array with its own set of square brackets:

int[][] numbers = { {1, 4, 2}, {3, 6, 8} };

Good to know: The int[][] specifies that the integer array is two-dimensional. A three-dimensional array would have three square brackets: [][][].

Access Elements of a 2D Array

Example

int[][] numbers = { {1, 4, 2}, {3, 6, 8} };

System.out.println(numbers[0][2]); // Outputs 2

Remember that: Array indexes start with 0: [0] is the first element. [1] is the second element, etc.

Change Elements of a 2D Array

You can also change the value of an element.

The following example will change the value of the element in the first row (0) and first column (0):

Example (this is not involved in ConsoleApp1)

```
int[][] numbers = { {1, 4, 2}, {3, 6, 8} };
numbers[0][0] = 5; // Change value to 5
System.out.println(numbers[0][0]); // Outputs 5 instead of 1
```

Loop Through a 2D Array

You can easily loop through the elements of a two-dimensional array with an enhanced for loop:

Example

```
int[][] numbers = { {1, 4, 2}, {3, 6, 8} };

for (int[] innerNumbers : numbers)
{
      for (int number : innerNumbers)
      {
            System.out.println(number);
     }
}
```

You can also use a for loop. Note that we have to use the length property of a two-dimensional array to specify how many times the loop should run:

Example (Row-Major Order)

```
int[][] numbers = { {1, 4, 2}, {3, 6, 8} };

for (int i = 0; i < numbers.length; i++)
{
      for (int j = 0; j < numbers[i].length; j++)
      {
            System.out.println(numbers[i][j]);
      }
}
```

Example (Column-Major Order)

```
int[][] numbers = { {1, 4, 2}, {3, 6, 8} };

for (int i = 0; i < numbers[0].length; i++)
{
      for (int j = 0; j < numbers.length; j++)
      {
            System.out.println(numbers[j][i]);
      }
}
```