Practice Assessment 2: ForEachIn2D

Goal
The goal of this code is to create a 2D array, assign values to its elements based on their row and column positions, and display the array. It also calculates and prints the total sum of all elements using a for-each loop.

How It Works

1. This part will store the value of the sum to zero, create a 2D array having a row of 3 and a column of 2. This will also print how many rows the array has.

```
int sum = 0;
int[][] nums = new int[3][2]; //This creates a 2D array; Row = 3, Column = 2

    // print the number of rows
    System.out.println("Length of rows: " + nums.length); //This will print how many rows the
array has.
                                                    //nums.length count how many rows in the array.
```

ROWS	COLUMNS	
0	0	0
1	0	0
2	0	0

2. This code checks if the 2D array has at least one row before accessing it to avoid errors. If there is at least one row, it prints the number of columns in the first row; otherwise, it prints that the array is empty.

```
 if (nums.length > 0) {
        System.out.println("Length of columns: " + nums[0].length); //This will access the length
of the first row (columns).
                                                    //nums[0].length counts how many columns in the
first row.
    } else {
        System.out.println("Array is empty, no columns.");
    }
```

3. This loop goes through each row and column of the array. It fills each element with the value of (row + 1) * (col + 1).

```
    for(int row = 0; row < nums.length; row++) {.
        for(int col = 0; col < nums[row].length; col++) {
```

```
            nums[row][col] = (row + 1) * (col + 1);
        }
    }
```

ROWS	COLUMNS	(row + 1) * (col + 1)	VALUE
0	0	(0 + 1) * (0 + 1)	1
0	1	(0 + 1) * (1 + 1)	2
1	0	(1 + 1) * (0 + 1)	2
1	1	(1 + 1) * (1 + 1)	4
2	0	(2 + 1) * (0 + 1)	3
2	1	(2 + 1) * (1 + 1)	6

4. This part of the code uses a for-each loop to go through each row and each element in the 2D array, printing all the numbers in a grid format. At the same time, it adds every element to sum and finally prints the total sum of all numbers.

```
    for(int[] rvals : nums) {
        for(int cvals : rvals) {
            System.out.print(cvals + " ");
            sum += cvals;
        }
        System.out.println();
    }
    System.out.println("Summation: " + sum);
  }
}
```