Worksheet: 2D Array Practice 1

1. Write the statement that will declare a variable and initialize it to **an array**:

a)	of ten `doubles`, all initialized to `0.0`	`double[] arr = new double[10];`
b)	of `String` array containing the names of three fruit	`String[] arr = { "peach", "pear", "cherry" };`

2. Write the statement that will declare a variable and initialize it to **a two-dimensional array**:

a)	of `double` with ten rows and twelve columns.	`double[][] arr = new double[10][12];`
b)	of `String` with five rows and four columns.	`String[] arr = new String[5][4];`
c)	of `int`, with three rows and four columns, initialized each array value using literal values equal to the sum of the row and column (`row+column`).	`int[][] arr = { { 0, 1, 2, 3 }, { 1, 2, 3, 4 }, { 2, 3, 4, 5 } };`

3. Write nested **enhanced `for`** loops to print the values in the array: each row on a new line, with each value in a row separated by a tab character.

```
1       for(double[] row: table) {
2           for(double value: row) {
3               System.out.println(value + "\t");
4           }
5           System.out.println();
6       }
```

4. Write a **method** named `sumTable` that takes a two-dimensional array of `double` named `table` and uses nested **enhanced `for`** loops to sum the values in the array.

```
1  public static double sumTable(double[][] table) {
2      double sum = 0.0;
3      for(double[] row: table) {
4          for(double value: row) {
5              sum += value;
6          }
7      }
8      return sum;
9  }
```

Worksheet: 2D Array Practice 1

5. Write a **method** named `tableMaxValue` that takes a two-dimensional array of `double` named `table` and uses nested **enhanced `for`** loops to find and return the maximum values in the array.

```
1  public static double tableMaxValue(
2                                  double[][] table) {
3      double max = table[0][0];
4      for(double[] row: table) {
5          for(double value: row) {
6              if(value > max) {
7                  max = value;
8              }
9          }
10     }
11     return max;
12 }
```

6. Write a **method** named `tableContains` that takes two parameters: `table`, a two-dimensional array of `String`, and `value`, of type `String`, which contains the value to find in the table. The method is to return `true` if a string equal to `value` is found in the array, otherwise `false`.

```
1  public static boolean tableContains(
2                  String[][] table, String value) {
3      for(String[] row: table) {
4          for(String cell: row) {
5              if(cell != null &&
6                              cell.equals(value)) {
7                  return true;
8              }
9          }
10     }
11     return false;
12 }
```