Objectives:
Practice using 2D arrays to store and to process values of different types.

A. Simple 2D array example

```java
public class TwoDArray {
    public static void main (String[] args) {
        int[][] table = new int[5][10];
        // Load the table with values
        for (int row=0; row < table.length; row++)
            for (int col=0; col < table[row].length; col++)
                table[row][col] = row * 10 + col;
        // Print the table
        for (int row=0; row < table.length; row++)
            for (int col=0; col < table[row].length; col++)
                System.out.print (table[row][col] + "	");
        System.out.println();
    }
}
```

1. Run this program and observe what it does.

2. The output produced is shown below. Circle the entries for `table[0][5]` and for `table[3][2]`

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
</tbody>
</table>

3. Modify the program so that the output rows and columns are labeled. The output should now look EXACTLY like this:

<table>
<thead>
<tr>
<th>#</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
</tr>
</tbody>
</table>
4. Modify the dimensions of the array – make it 3 rows by 4 columns and run the program again. The output should still look OK without having to change anything else in the program. If necessary, adapt your program so that it works with any reasonable dimensions (note that there is a limit to how many columns can be displayed across one line, so it is not expected to work well with large values for the number of columns).

B. 2D array of double

1. Make a new version of your program from part A that creates an 2D array of 5x5 values of type double, set to random values in the range 0…1 (use Math.random()).

2. Examine your code: you should still have two nested loops (one to initialize the 2D array and one to print its contents). Add a third nested loop to process the array entries one more time, this time counting how many are greater than 0.5 (since the entries were generated randomly, it should be about half of them), and some code to display the count. Run the program a few times to observe the counts:

3. Increase the array size to a 100x100 array and try again:

4. (optional) Incorporate a timer in your program. How much longer does it take to run for different size arrays? (You can try 10x10, 10x100, 10x1000, 10x10000, etc and observe the times in milliseconds.)

C. 2D array of boolean

1. Make a new version of your program that creates a 2D array of 5x5 values of type boolean. Use an initializer list to instantiate and initialize the array table to represent the following configuration, where the asterisk indicates a true value at that position in the array (no asterisk means false). For example, table[1][3] is false, whereas table[2][1] is true.

2. Suppose indices represent people and that the value at row i, column j of this 2D array is true just in case i likes j, and false otherwise. For example, person 3 likes persons 0, 1, and 2, but does not like person 3(self). Write some code to count how many possible matches are represented in the array. Note: match means that person x likes person y and vice-versa; in the above example we have two matches: (0,1) and (1,2).
Lab Comments

Comments on this lab, please:

What was the most valuable thing you learned in this lab?

What did you like best about this lab?

Was there any particular problem?

Do you have any suggestions for improving this lab as an effective learning experience?