Figure 12.2 Two-dimensional arrays.

Suppose we have a table of marks:

	quiz #1	*quiz #2*	*asgn #0*	*quiz #3*	*mid term*
Cathy	8	7	9	8	24
Richard	7	7	8	7	22
Homer	6	8	8	6	17
Lisa	9	9	9	9	27

This table is inherently two-dimensional, with a *row* for each student and a *column* for each mark. For example,

- Lisa's quiz #2 mark is in Lisa's row (row 3) and quiz #2's column (column 1).
- Richard's mid term mark is in row 1, column 4.

We think of Lisa's quiz #2 mark and Richard's mid term mark as being elements $(3, 1)$ and $(1, 4)$ of the table, respectively.

In Java, we naturally represent a table like this as a two-dimensional array of **int**. We would declare it as:

$\boxed{\quad\quad\quad\quad}$ **table;**

instantiate is as:

table = new $\boxed{\quad\quad\quad\quad}$;

and read in its values as:

for (int row = 0; row < table.length; row++)
for (int col = 0; $\boxed{\quad\quad\quad\quad\quad\quad}$ **; col++)**
table[row][col] = Integer.parseInt(keyb.readLine());

This is an example of a *doubly-nested* **for**-loop. The elements of **table** are input in the following order:

Exercise: Declare and instantiate a *three*-dimensional array of **int**, and input its values using a *triply-nested* **for**-loop. Be sure that you understand the order in which the elements are being input!