July 9, 2010

2D Arrays (Multidimensional Arrays)

Let's write a program to print the 12 by 12 multiplication tables using two 'for' loops.

```
for (int i = 1; i <= 12; i++) {

    for (int j = 1; j <= 12; j++)

        System.out.printf("%3d ", i*j);

    System.out.println("");

}
```

Let's write the numbers from 0 to 143 using two 'for' loops.

```
for (int i = 0; i < 12; i++) {

    for (int j = 0; j < 12; j++)

        System.out.printf("%3d ", i*12+j);

    System.out.println("");

}
```

Let us look at the declaration of an array of values:

```
datatype[] name = new datatype[size];
```

- datatype → Actual data type of the array
- name → Name of the variable used to identify the array
- new → Java keyword to declare a new block of memory
- size → A positive integer which represent length of the array

Let us look at the declaration of a 2D array of values:

```
datatype[][] name = new datatype[first_size][second_size];
```

- first_size → Length of first array dimension (Think of this as "height.")
- second_size → Length of the second array dimension (Think of this as "width.")

Let's look at the declaration for an N-D array of values:

```
datatype[][]...[] name = new datatype[first_size][second_size]....[N_size];
```

In the N-D array, you need 1 blank "[]" for each dimension on the left side of the equals. On the right side of the equals, you need the same number of "[]". At least one of these "[]" blocks must be defined with a size.

```
int height = 2;

int width = 3;

int[][] array = new int[height][width];


int[0][0] = 1;

int[0][1] = 2;

int[0][2] = 3;

int[1][0] = 4;

int[1][1] = 5;

int[1][2] = 6;
```

This will result in an array looks like this:

1 2 3

4 5 6

We can also declare a 2D array by directly assigning data to the array when you declare the array:

int[][] baseBallScores = {{0,0,0,0,1,0,0,0,0,3,1,0},{0,0,0,3,0,0,0,1,0,4,4,1}};

This results in this array:

0 0 0 0 1 0 0 0 0 3 1 0

0 0 3 0 0 0 1 0 4 4 1

To index through a 2D array and print the values in an array named "array":

```
for (int i = 0; i < array.length; i++) { // Index though each row

    for (int j = 0; j < array[i].length; j++) { // Index though each column

        // Do something with value at array[i][j]

    }

}
```

2D Matrix as an Object

```
class Matrix<T> {

    public T[][] a;

    protected int height;

    protected int width;


    MatrixE(int height, int width) {

        this.height = height;

        this.width = width;

        a = (T[][]) new Object[height][width];

    }


    public String toString() {

        String s = "";

        for (int i = 0; i < a.length; i++) {

            for (int j = 0; j < a[i].length; j++) {

                s += a[i][j] + " ";

            }

            s += "\n";

        }

        return s;

    }


    void swapRow(int p, int q) {

        if (p < 0 || p >= height || q < 0 || q >= height) {
```

```java
                System.out.println("Error: Out of bounds on attempt to swap rows.");
                System.exit(1);
            }


            T temp;
            for (int i = 0; i < width; i++) {
                temp = a[p][i];
                a[p][i] = a[q][i];
                a[q][i] = temp;
            }
        }


        void swapCol(int p, int q) {
            if (p < 0 || p >= width || q < 0 || q >= width) {
                System.out.println("Error: Out of bounds on attempt to swap columns.");
                System.exit(1);
            }


            T temp;
            for (int i = 0; i < height; i++) {
                temp = a[i][p];
                a[i][p] = a[i][q];
                a[i][q] = temp;
            }
        }
    }
```