

Two-Dimensional Arrays

Presented by Dr. Irwin Pascual
Van Nuys Math/Science Magnet

Routine Conventions

Two-dimensional arrays are used to store data in situations where there is a useful purpose for incorporating a two-way logic for storing the data. The accompanying problem involving the Burger Blast fast food chain's sales information for a recent week's operations incorporates this sort of logic. The fast food chain has five sales locations. One of the dimensions allows the sales information from each location to be maintained separately. The data covers a seven-day week. The second dimension allows the sales data from each day of the week to be maintained separately.

The "visual" representation of an array contains a combination of rows and columns. Each row and column is numbered and the numbering convention in Java has the uppermost row being numbered as row number 0 and the leftmost column being numbered as column number zero. Each location in the array is designated by the combination of both its row and its column number. In mathematics this combination is written as an ordered pair with the row number being listed first and it being followed by the column number with a comma being inserted to separate the row and column designations. These locations are commonly referred to as cells and/or elements. Element 3, 4 would refer to the data located on row number 3 in column number 4. In Java [as well as in many other computer programming languages], prior to placing the data into computer memory the area of computer memory that will be used needs to be given a name. Once it is assigned a name it can then, at a later time, be assigned a size designation. Java programmers utilize a specific convention for assigning names. The first letter of the name is generally a "lower case" letter and the names are most often made up of compound words with the meaningful portions of each compound name following the first portion beginning with an "upper case" letter. The names are not allowed to contain "blank" spaces. The array being used in the accompanying problem will be designated using the name **salesData**. If we were interested in identifying the information contained on row number 3 in column number 4 we would utilize the following code: **salesData[3][4]**. If this were a mathematics problem instead of a computer program we would have used **salesData (3,4)**.

Part II of the problem contains the sales data. There are five rows of data. Each row represents one of the five retail locations. As well, there are seven columns of numbers. Each represents a different day of the week. Neither the actual computer nor the Java programming language recognizes the differences between retail fast food store locations or between different days of the week. As well, neither understands the concept of food or money. These are built into the user's interpretation of the computer program. Comments inserted into the computer code, statements and instructions displayed while the program is operating and the selection of meaningfully descriptive names to use in the identification of the data (attributes) and of the sections of instructions commonly referred to as methods are utilized to communicate these concepts from person to person. In this problem the rows are numbered consecutively as rows 0, 1, 2, 3 and 4 while the columns are numbered as columns 0, 1, 2, 3, 4, 5 and 6. Somewhere in the comments and/or printed instructions a program such as the one that would be written to accomplish the task being presented today, there would be an indication of which of the five rows of data being stored would correspond to each of the five existing retail sales locations. As well, the column that referred to each of the seven days of the week would also be identified.

Traversing a Two-Dimensional Array

A “traversal” refers to the action of moving through the section of memory storage containing the locations for the data. Our five-row and seven-column two-dimensional array will have thirty-five compartments (i.e., cells or elements). Each compartment will be the same size. Each will contain a sufficient number of bytes to store the largest amount of information that could possibly need to be stored at that location. Determining the total number of compartments necessary will enable Java to reserve a block of consecutively located addresses in the computer workstation’s RAM to store all of the data involved in the problem. A systematically organized algorithm can be created so that each compartment can be “visited” in an efficient fashion. Each time each compartment is visited the computer program’s instructions can perform tasks such as storing data, modifying the data, evaluating the data, performing arithmetic operations with the data, displaying the data and so forth. A very commonly used technique involving an organized and systematic traversal of a two-dimensional array instructs the computer to begin with the uppermost row [row number 0] and proceed from left to right moving from the leftmost column [column number 0] to the rightmost column [the column number that would correspond to the number one less than the total number of columns contained in the array] in a sequential succession while also dropping down to the next row number once every column in the current row had been visited. This would continue until the computer program had sequentially visited every row up to and including the row number designated by one less than the total number of rows contained in the array.

The algorithm governing the traversal of the two-dimensional array is accomplished through the use of a “looping” technique. This technique utilizes a block of code that is repeatedly executed. The code is written in a fashion that allows the row and column numbers to systematically change in controlled increments that incorporate a pattern that results in each compartment being visited once and none of the compartments being accidentally skipped.

The activities involved in creating this code are organized in a fashion that reinforce the utilization of programming techniques which insure that the size of each dimension of the array are established prior to the beginning of the computer program’s first traversal of an array.

An array is generally one of the attributes of a class and the coding technique utilized allows for it to be given a “name” without having its size established prior to the point in time when the code for the class is compiled.

The actual number of rows and columns involved in an array is established subsequent to the naming of an object being created either prior to or simultaneous to the point in time when data will be entered into the computer’s RAM.

Programming code that directs the computer to utilize the data contained in the array is executed after data has been entered into RAM.

Since the size of each dimension is pre-established prior to the inception of the first looping technique, coding techniques involving a pre-established number of loops and organization that governs the sequential changes in the compartment (cells/elements) identification can be utilized. These coding techniques enhance the versatility of the methods they are a part of and simplify programming efficiency.

An Example

The following segment of code will illustrate the display of the data that is involved in our Burger Blast problem:

```
for (int store = 0; store < numberOfStores; store++) {
    for (int day = 0; day < numberOfDaysInWeek; day++) {
        System.out.print(salesData[store][day] + " ");
    }
    System.out.println("");
}
```

In this example since there are five rows (for the stores), the row numbers would progress from 0 to 4. Since there are seven columns (for the days of the week), the column numbers would progress from 0 to 6. Two sets of looping instructions are involved in the example provided above. There is an inner loop that is described as being “nested” within what is described as an outer loop. In this case the inner loop uses the integer variable “day” to keep track of the column numbers and the outer loop uses the integer variable “store” to keep track of the row numbers. Executing the first line establishes a value of 0 for store. That value does not change until the next line is executed once for each of the column number designations. The line containing the array location referred to as **salesData[store][day]**, will end up being executed once for each of the different compartment locations in the array. When there are five rows and seven columns it will be executed thirty-five times [5 x 7 = 35]. The systematic traversal algorithm will allow the task of having the line executed once for each compartment to be successfully accomplished. The line utilizing the derivation of the print command containing the instruction “**println**” will accomplish the task of separating each row into a separate line of output. Within the command containing the instruction “**print**” there is a block of blank spaces being printed. This will enable the data for each separate day of the week to be distinctively separated from that for the day either immediately prior or after the day being printed.

Burger Blast
Weekly Sales Records



Part I: Problem Statement

Burger Blast is a new fast food chain that has opened up five locations in the greater Los Angeles metropolitan area. They serve only a value meal consisting of a quarter pound hamburger, an order of fries and a standard sized soda. Their meal costs \$3.00 including tax. All five locations are company owned and supervised. Their operations manager, I.M. Anut, has asked his Programming Department to devise a computer program that can be used to analyze their restaurants' sales figures so he can determine which locations require additional staffing and on which days this staffing is needed. As well, he wants to be able to determine which locations could handle their current sales activities with fewer staff.

The sales information will be stored in a 5-row by 7-column two-dimensional array.

Each row will represent one of their greater Los Angeles restaurant locations and each column will represent a different day of the week, beginning with Sunday.

Part II: Data Existing In Computer Memory

The following data, containing location-by-location, day-by-day sales information has already been stored in computer memory. Each entry represents 100 value meals.

5	7	3	8	9	2	4
6	2	5	8	3	5	2
3	7	2	9	3	5	7
7	4	7	7	3	4	2
2	6	5	8	2	4	3

Part III: The SalesAnalysis Class

The following section contains some of the class attributes, constructors and methods that are involved in this problem. You will be required to write out the code for two of the methods listed on this page.

```
public class SalesAnalysis {

    private int numberOfStores;
    private int numberOfDaysInWeek;
    private int[][] salesData;

    public SalesAnalysis() { }

    public SalesAnalysis(int ns, int nd, int[][]temp) {
        numberOfStores = ns;
        numberOfDaysInWeek = nd;
        salesData = temp;
    }

    public int getBestSalesDay() {
        // code for this method needs to be written
    }

    public int getBestSalesLocation() {
        // code for this method needs to be written
    }

    public int getWorstSalesDay() {
        // code for this method needs to be written
    }

    public int getWorstSalesLocation() {
        // code for this method needs to be written
    }

    public void DisplaySalesData() {
        for (int store = 0; store < numberOfStores; store++) {
            for (int day = 0; day < numberOfDaysInWeek; day++) {
                System.out.print(salesData[store][day] + " ");
            }
            System.out.println("");
        }
    }
}
```

Part IV: Your Assignment

The “**accessor**” methods involved in this problem need to be written. Write the code for at least these two: **getBestSalesLocation()** and **getWorstSalesDay()**.

```
public int getBestSalesLocation() {
```

```
}
```

```
public int getWorstSalesDay() {
```

```
}
```