2.4.3 Multidimensional Arrays

A *multidimensional array* is an array that is accessed by more than one index.

Sometimes arrays need to be accessed based on more than one index. A common example of this is a matrix. A *multidimensional array* is an array that is accessed by more than one index. It is allocated by specifying the size of its indices, and each element is accessed by placing each index in its own pair of brackets. As an example, the declaration

```
int [ ][ ] x = new int[ 2 ][ 3 ];
```

defines the two-dimensional array `x`, with the first index ranging from 0 to 1 and the second index ranging from 0 to 2 (for a total of six objects). The compiler sets aside six memory locations for these objects.

2.4.4 Command-line Arguments

Command-line arguments are available by examining the parameter to `main`.

Command-line arguments are available by examining the parameter to `main`. The array of strings represents the additional command-line arguments. For instance, when the program is invoked,

```
java Echo this that
```

`args[0]` references the `String` `"this"` and `args[1]` references the `String` `"that"`. Thus the program in Figure 2.5 implements the `echo` command.

```
 1 public class Echo
 2 {
 3     // List the command-line arguments
 4     public static void main( String [ ] args )
 5     {
 6         for( int i = 0; i < args.length - 1; i++ )
 7             System.out.print( args[ i ] + " " );
 8         if( args.length != 0 )
 9             System.out.println( args[ args.length - 1 ] );
10         else
11             System.out.println( "No arguments to echo" );
12     }
13 }
```

Figure 2.5 The `echo` command