

Arrays and ArrayLists

By Edward Goldthorpe, AJ Piergiovanni, Brian McCormack

Arrays:

Arrays are essentially a simple way to store lists of data of the same type, together whether they are primitives like ints, or objects, which could be anything. You could even have an array of arrays. They are fixed in length, so make sure you choose the correct length. You can declare an array like this `Datatype[] arrayname = new array[number of elements]` The array's index numbers from zero to its length -1, making index out of bounds errors common due to attempting to access the final element using the length. To access a specific element or put a value in, we type `arrayname[element number]` (for retrieving value) or `arrayname[element number] = value` (for inserting value). The easiest way to construct a filled array without having to choose the length or enter in individual values is by typing the array, then a list of values. Note that once you create the array, the number of values is set. `Datatype[] arrayname = { value1, value2...}` Besides the elements in an array, the only other really useful feature of an array is its length field, which provides the array length. When the user sets an array length, it cannot be changed. You can copy and expand an array by using the method the `Arrays.copyOf(arrayname, new length)` method, but be careful if you do so, as this can have its own problems.

2D Arrays:

The most common use of 2D arrays is to represent a grid. For example, if you wanted to make a chess game, you could use a 2D array of integers to represent the board. Each chess piece would be represented by a different int.

```
public static void main(String[] args) {  
    final int EMPTY = 0;  
    final int PAWN = 1;  
    final int KING = 2;  
    final int QUEEN = 3;  
    final int KNIGHT = 4;  
    final int ROOK = 5;  
    final int BISHOP = 6;  
  
    int board[][] = new int[8][8]; // This is the important line.  
    // it creates the 2D array. Each set of brackets added after the  
    // variable name  
    // creates another dimension to the array. Essentially, a 2D array is  
    // nothing more  
    // than an array of arrays that contains an object (int's in the  
    // example).  
    // The new int[8][8] creates the array with 8 arrays that each hold 8  
    // int (because a  
    // chess board is 8x8.  
  
    // To access a value:  
    board[0][0] = ROOK;  
    board[1][0] = PAWN;  
    // etc.  
  
    // If you wanted to access every member of the array and do something to  
    // it, for example, clear the board, you do that like this:  
    for (int i = 0; i < 8; i++) {  
        for (int j = 0; j < 8; j++) {  
            board[i][j] = EMPTY;  
        }  
    }  
}
```

ArrayLists:

An ArrayList is similar to an Array in that it serves a similar function: to hold a number of objects. However, once an Array has been created, the length of the Array cannot be changed. An ArrayList, on the other hand, can have objects added to it any number of times. For example, an Array instantiated with 5 positions could not have a 6th added without a new Array, but an ArrayList could have any number of extra positions added or removed as necessary. Additionally, ArrayList is a Java class, with methods being used to modify or retrieve information from the ArrayList.

Example:

```
//Creates an ArrayList list
ArrayList list=new ArrayList();
//Adds a string, an int, and a double to list
list.add(2);
list.add("a");
list.add(3.14);
System.out.println(list);

//Adds another string to list, extending the ArrayList
list.add("hello");
System.out.println(list);

//Removes the second object from list
list.remove(1);
System.out.println(list);
```