06.01 Virtual Lecture Notes (Part 2)

Numeric Array Basics

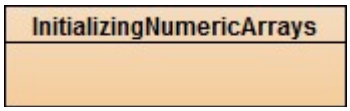
The syntax for declaring and initializing numeric arrays is the same as for String arrays.

```
int [] intValues;
intValues = new int[10];

or

int maxIndex = 100;
double [] doubleValues = new double[maxIndex];
```

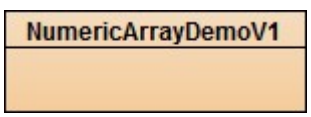
The number of positions in an array, also known as the length, can be assigned with an actual value (e.g., 10) or with a variable (e.g., **`maxIndex`**). The context of each program will indicate whether to use values or variables, and whether to declare and initialize an array with one statement or two. Before anything is assigned to an array, each index position is initialized with a default value of zero, indicating it is empty.



Run the InitializingNumericArrays program. Notice how the numeric array is declared. When the array is traversed to print the contents of each index position, you'll see the default values displayed.

Convert the array to hold double values. What do you predict will be the initial value of the index positions? Once a numeric array is declared and initialized, values can be assigned to index positions using the same techniques you learned about with **`String`** arrays.

Direct Assignment within the Source Code



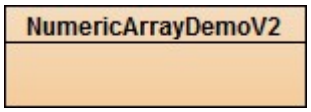
Run the class called NumericArrayDemoV1. An integer array of length 5 is declared and assigned values. Initially, the values are assigned after declaration of the array.

Modify the code by commenting and uncommenting sections to view how to directly assign values during initialization.

```
int [] numList = {10, 9, 8, 7, 6};
```

In both cases, the array is traversed to print the value of each index position. Now, try to convert the program to work with **`doubles`**. What happens if you attempt to assign an int to a double array and vice versa? Continue to experiment with the program on your own.

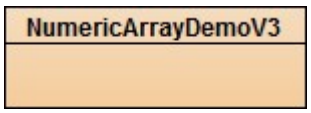
User Assignment from the Keyboard



Run the class called NumericArrayDemoV2 and provide numeric values as prompted. The program assigns the user input provided via keyboard to each individual index position of the array. Then it prints the value of each index position by traversing the array.

Try to change the program so it will work with **`int`**. Experiment with the program on your own. What happens if you try to assign **`double`** to an **`int`** array and vice versa?

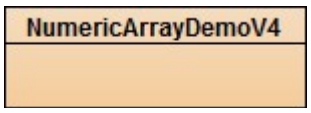
Read Array Data from a File



Run the class called NumericArrayDemoV3. This program will read the numbers in the **`doubles.txt`** file into the array. Then prints the value of each index position. Observe the output. Once it works for doubles, change the program so that it will read the **`ints.txt`** file. Experiment with the program on your own.

Notice how the first while loop is used to determine how many values are contained within the text file. Once that count is known, the array can be declared at the appropriate length.

Finding the Sum of Array Elements



One of the most common programming tasks is calculating the sum of the elements in an array. Open and run the NumericArrayDemoV4 program. Notice the code used to traverse the array and add each value to the accumulating total.

In earlier programs, simple variables that could only hold one value at a time were used, so you calculated sums by adding values as they were read in from a file or entered from the keyboard. With arrays, tasks can be broken down into smaller tasks because all of the data is simultaneously stored in the memory. Consequently, the array can be traversed from beginning to end, adding each element to the sum.

Finding the Maximum and Minimum Values in an Array

Another common array processing task is to determine the maximum and minimum values of an array. Again, use the NumericArrayDemoV4 program as a guide.

```
int [] intValues = {5, 6, -12, 73, 4, -25, 15, 8, 0, -2};
```

By visually scanning the array in the previous example, you can instantly tell that 73 is the maximum value and -25 is the minimum value. But how can you write the Java code to select the maximum and minimum values?

Pretend you are the computer, traversing the array from beginning to end. What decisions do you have to make in order to identify the largest number and the smallest number? What do you do with a number when it is the maximum or minimum? Can you translate your ideas into code? Compare your idea to the one provided in the demonstration program. Make changes to make your ideas work.

Print