

1581 Laboratory 6

Arrays

Jonathan Redmann

Introduction

In this lab we will be working with arrays. Arrays are an example of what are called ***data structures***. Data structures are programming techniques for organizing data in a way that makes accessing the data fast or working with the data easier in some way. In this lab we will see how to use arrays effectively.

Declaring and Creating Arrays

First we need to understand the different ways of creating arrays. We can either declare an empty array of a specific size or we can declare a non-empty array supplying a specific list of values to store. In the second case the computer infers the size of the array from how many things are in the list of initial values.

The syntax for array creation is:

```
dataType[] arrayName = new dataType[ arraySize ];
```

for example

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

creates an array that can hold 10 int type values. The array's name is exampleArray.

Or we can create an array like this:

```
dataType arrayName = { valueOne, valueTwo, valueThree };
```

for example:

```
String[] weekDays = { "Mon", "Tues", "Wed", "Thurs", "Fri" };
```

EXERCISE ONE: Lets create two arrays.

1. Open Gedit and type in the "boiler-plate" code.
2. In the main method create an array of 10 integers using the first method.
3. Assign values to the 10 elements of the array using a class Random object to get random integers.

4. Write a for loop that prints the values in the array.
5. Now declare an array of 12 Strings using the second method. The values will be the names of the 12 months.
6. Use the random number generator from step 3 to randomly select an element of the String array and print that element.

Using Arrays

Now that we have some experience creating and accessing the elements of arrays lets look at some uses. A very common uses of arrays is lookup tables that allow us to translate from one type of information to another. Compilers, like the Java compiler use a type of lookup table called a ***symbol table*** to convert variables to the corresponding memory address where the variable is actual stored in the hardware.

Another common use of arrays is to represent vectors and matrices from mathematics. The branch of mathematics that uses these is called ***linear algebra***. Linear algebra is used in computer graphics, cryptography, artificial intelligence, and simulations, among many other things. It is perhaps the most widely used field of applied mathematics.

EXERCISE TWO: (Binary Lookup Table) Translation Tables. Lets create a table that allows us to quickly and easily convert between decimal and binary numbers. If you are not familiar with binary numbers, don't worry, below is a table of binary numbers and their decimal equivalent:

Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Use the table above to create an array that will allow us to lookup the binary representation of the integers 0 to 8.

Note: Since numerical data types will remove unnecessary leading zeros, use Strings to store the binary version of each number.

1. Open a new file in Gedit and type in the “boiler-plate” code, import the Scanner.
2. In the main method use the second type of array declaration to create an array of Strings containing the binary representation of the numbers 0 to 8.
3. Initialize the Scanner object for input.
4. Write a loop to ask the user for a number between 0 and 8.
5. Inside the loop use the array to “lookup” the binary representation and print it to the screen saying “the number ” number_given “ is “ binaryVersion “ in binary”.
6. Save, add, commit, compile, and run your program.

EXERCISE THREE: (Vector Multiplication) Linear Algebra. Lets now write a program to calculate the dot product of two vectors. We will use arrays to represent the vectors.

A vector is just a list of numbers where the order of the numbers matters. The general idea and use of vectors is beyond the scope of this lab, so we will just cover the process of multiplying two vectors with the dot product.

In order to take the dot product of two vectors, the vectors have to be the same length. Lets look at an example:

$$v_1 = \langle 1, 2, 3 \rangle \text{ and } v_2 = \langle 4, 5, 6 \rangle$$

then
$$v_1 \cdot v_2 = 1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6$$

In other word the to take the dot product multiply the first element of each vector together and add that to the result of multiplying the second terms of each vector, and so on.

1. Open a new file in Gedit and type in the “boiler-plate” code, including importing the Scanner.

2. In the main initialize the scanner object
3. Declare two int arrays each of size 3.
4. Write a loop asking the user for the six values needed for the two arrays. Store the values into the arrays directly.
5. Declare a separate method for calculating the dot product. It should take two arguments; the two arrays representing the vectors.
6. In the dot product method write a loop that will iterate over the elements of the two arrays, first multiplying them together and then adding the result to a sum variable.
7. return the resulting sum.
8. In the main method call the dot product method passing it the two arrays and print the value returned from the method.

Using Command Line Arguments

As you have seen with the commands you learned for the Linux command line, arguments are an simple way to supply input and to provide options that change the action performed by the command. Adding command line arguments to your programs is not difficult.

EXERCISE FIVE: In this exercise we will modify our vector dot product program to accept the values of the two vectors as command line arguments.

1. Open ExerciseThree.java and save it as ExerciseFive.java be sure to change the class name to ExerciseFive.
2. In the declaration of the main method the main method is receiving a parameter called “args” which is a String array. This is array holds any command line arguments. First we need to check that there is the correct number of arguments and if there is not, print an error message. Write a conditional that checks the length property of the args array to see if it has 6 arguments.

3. Since each vector takes 3 numbers, indices 0 to 5 should be the values for the two vectors. Write a loop that assigns the first 3 arguments to the first vector, and the last three arguments to the second vector.
4. The rest of the program should remain the same.
5. Save, add, commit, compile, and run.

Class ArrayList

Class ArrayList provides objects that work much like regular arrays but their size is not fixed, unlike regular arrays. ArrayLists also provide methods for standard operations we would want to do such as adding and element, removing an element, getting the size of the list, and so on. To see a full list of the methods available go to:

<http://docs.oracle.com/javase/6/docs/api/java/util/ArrayList.html>

This is an important note here. All of the classes supplied by Java have documentation available online. This documentation is referred to as the Java API documentation. You can find the details on any class in the API by Google searching “Java API” and the class name. You need to get familiar with the API documentation as you will be using it extensively for the next few semesters.

EXERCISE SIX: (Student Roster) Sorting and Searching. Let us now see how to use ArrayLists. Lets create a student roster. We will use an ArrayList of Strings to store the names of the students and provide basic roster functions like adding students to the roster, removing students, getting the total number of students, and determining if a specific person is enrolled in the class.

1. Open a new file in Gedit and type in the “boiler-plate” code. Be sure to import the Scanner class.
2. Write a menu method that will ask the user to chose from: 1) Add a student, 2) Remove a student, 3) Check enrollment of a student, 4) Print the roster, 5) Get the number of enrolled students, or 6) Quit. Be sure to have your menu method return an int representing the choice the user made.
3. In the main method initialize an ArrayList object.
4. Write a loop that exits when the user choose quit.

5. Inside the loop write a switch statement that processes the users choice.
6. Inside some of the cases you will need to add print statements and get input from the user. You will also need to call the appropriate method of ArrayList to perform the operation the user chose.
7. Save, add, commit, compile, and run your program.

Just for Fun

Some of you may be wondering why arrays are indexed starting at zero. If you want the answer, read this explanation from Edsger Dijkstra, one of the people involved in establishing this convention:

<http://www.cs.utexas.edu/users/EWD/ewd08xx/EWD831.PDF>