Lab Workbook 6

Software Development HDSWD
Arrays
17/06/12

Jonathan Meaney

For these problems you will be building up a single class with useful methods that can be performed on arrays. The name of this class is ArrayUtilities. Create a class called ArrayUtilities and a main class to test it out. The ArrayUtilities class has no data members or constructors, we're only interested in the methods it can perform. Create an object of the ArrayUtilities class to run the methods in the main class. You can use the following array as a starting point. Try change the numbers to see different outputs from the methods.

int[] numbers = {1,4,13,43,25,6,44,9,7,26,76,45,12,9,5,3,2,5,12};

You can create an object of the ArrayUtilities class like any other class:

ArrayUtilities utility = new ArrayUtilities();

Then you can run any methods available to the object and if it is returning something store it in an appropriate variable. Remember when passing an array to a method don't forget the square brackets []

int sum = utility.sumArray(numbers[]);

Problem 1:

Create a method in the ArrayUtilities class called sumArray which accepts an array of integers and returns the sum of the integers in the array. Pass the numbers array into the method in the main class. Use a variable to store what is returned and print it out.

Problem 2:

Create a method in the ArrayUtilities class called averageArray which accepts an array of integers and returns the average of the integers in the array. Pass the numbers array into the method in the main class. Use a variable to store what is returned and print it out.

Problem 3:

Create a method in the ArrayUtilities class called maxArray which accepts an array of integers and returns the maximum value found in the array. Pass the numbers array into the method in the main class. Use a variable to store what is returned and print it out.

Problem 4:

Create a method in the ArrayUtilities class called minArray which accepts an array of integers and returns the minimum value found in the array. Pass the numbers array into the method in the main class. Use a variable to store what is returned and print it out.

Problem 5:

Create a method in the ArrayUtilities class called printArray which accepts an array of integers and prints out all its values on one line separated by spaces.

Problem 6:

Create a method in the ArrayUtilities class called copyArrays which accepts two integer arrays. The method will copy all of the elements from the first array and put them in the same order in the second array. Create a second integer array in the main class and pass it along with the numbers array into the method. Use the printArray method with both arrays before and after executing the copyArrays method.

Problem 7:

Update problems 1 – 4 to accept arrays of type double and return doubles. Remember you can have two methods of the same name as long as their parameter list is different, i.e. different datatypes.

Problem 8:

Update problem 5 to accept 2D arrays. Remember you can have two methods of the same name as long as their parameter list is different, i.e. different array dimensions.