CSC 241	*Java Data Structures*	Lab 4

PROJECT   Arrays and Methods with Java - Searching and Sorting

Objective   To construct a program that uses methods and arrays for searching and sorting.

PROJECT DESCRIPTION

Arrays, or subscripted variables, allow for variables, with common data types, to be grouped under one name. Methods assist in developing modular programs segments.

Type, compile and run a computer program that uses both arrays and methods with Searching and Sorting Techniques.

Follow the provided instructions and use arrays and methods to search for data and sort the data.

Basically, your program should perform these minimally required tasks:

- •   declare and populate an array of strings
- •   declare and populate a parallel array of integers
- •   perform searches on both arrays
- •   perform parallel processing using both arrays
- •   display the arrays after some elements in the arrays have been updated
- •   accomplish sorts on both arrays

These tasks will be outlined below in the steps for completion of this project.

After you complete your program code, compile and run your program, observe the output and then modify the program.

Some skeletal program code is shown in **Figure 1**, which follows.

Information About This Project

Searching techniques include the linear search routine and the binary search routine.

Linear Search

A Linear Search seeks an item or value in a list of elements. A linear or sequential search compares each element in the list with the element sought. If the item or value is found, the search ceases otherwise a message will indicate that the search was unsuccessful. A linear search of n items requires n comparisons in an attempt to locate the item sought. The average number of comparisons is typically $n \div 2$.

Binary Search

A Binary Search requires that the list is first placed in sorted order. Essentially, the binary search first compares the element sought with the middle value in the list. If the middle value is the element sought, then the search routine ends. If the middle value is less than the element we seek, then, in a divide and conquer fashion, we search the values or items below the middle value. Otherwise we look for the item in the values above the middle value.

Sorting techniques include the Exchange or Bubble Sort and the Insertion or Bully Sort.

Bubble Sort

The Bubble Sort takes a list and starts by comparing the first two elements and then determines the larger of the two. A swap is performed to place the lower element to be first in the list. In succession, we compare the next pair of values in the list, perform a swap of values and continue in this fashion until the list is sorted.

Insertion Sort

An Insertion Sort, also known as a Bully Sort, is such that we take the elements from the list one at a time and insert them in their correct position into a new sorted list.

PROJECT   Arrays and Methods with Java - Searching and Sorting

Steps to Complete This Project

STEP 1   **Open an Integrated Development Environment (IDE)**

Open Eclipse, Net Beans, MS Visual Studio, or similar programming text editor / compiler.

STEP 2   **Write the Program Code**

Write the program code that will satisfy the requirements of this project.

In a top - down fashion, construct the code statements with these specifications.

Within your file, declare a **Scanner** class object that will be used for user input.

```
import java.util.Scanner;

static Scanner sc = new Scanner(System.in);
```

- Declare an array of strings. Use a one - dimensional array with ten elements.

```
String[] myClientsArray = new String[10];
```

- Populate the array of strings using these client names.

        Client Names

        Butler
        Samuels
        Bond
        Chang
        Baker
        Davis
        Zheng
        Joe

Use this code construction for hard - coding the values of the string array.

```
myClientsArray[0] = "Butler";
myClientsArray[1] = "Samuels";
myClientsArray[2] = "Bond";
.
.
.
```

- Declare an array of integers. Use a one - dimensional array with at most ten elements.

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

- Populate the array of integers using these client numbers. This integer array will be a parallel array to the string array, which holds the client names.

PROJECT   Arrays and Methods with Java - Searching and Sorting

Client Numbers

108
121
188
107
122
111
203
135

Use this code construction for hard - coding the values of the integer array.

```
myClientNumbers[0] = 108;
myClientNumbers[1] = 121;
myClientNumbers[2] = 188;
.
.
.
```

- After populating the arrays, you will now search the **Client Names** list. Using a linear search method, such as that given below, implement the code to search for an element in the above list. In the **main()** method use a call to the **linSearchString()** method.

```java
public void linSearchString(String strArray[], String search)
{
    int i = 0;
    int flag = 0;
    for(i = 0; i < 10; i++)
    {
        if (search.equals(strArray[i]))
        {
            flag = 1;
            break;
        }
    }
    if (flag == 1)
      System.out.println("item found at position " + ( i + 1));
    else
      System.out.println("item not found");
}
```

- In a similar fashion, you will now search the **Client Numbers** list. Write and use a **linSearchInt()** method, whose code is to be comparable to the above method.

PROJECT   Arrays and Methods with Java - Searching and Sorting

STEP 3   **Build, Compile and Run the Program**

Compile and run your program code statements. Correct any syntax or compile errors.

STEP 4   **Test the Program**

Once you have successfully compiled your program, test your program and search for a client name and a client number. Observe the output in the **Console** window of your application.

STEP 5   **Verify Your Output**

When you enter the above information, the **Console** window should show your program output. Verify that the output is accurate and that the required output information is displayed. If necessary, use separate program runs to compare various input and output values.

STEP 6   **Modify Your Program**

With both arrays unsorted, use parallel array processing logic and programming techniques to search for a client name and have the associated client number also displayed in the search results.

Also, attempt to alter your current program such that it will have some routines to allow the user to change a particular client name or client number.

Now reassign the two original arrays that were named as **myClientsArray** and **myClientNumbers** into two new arrays, which use their original respective data types.

You will perform a sort on both these new arrays using this intrinsic sort routine.

Incorporate the following **import** statement, which will allow for an intrinsic sort method to be used.

```
import java.util.Arrays;
```

Below your existing code in the **main()** method, sort both string and integer arrays using this format.

```
Arrays.sort(array_name);
```

STEP 7   **Re - Compile and Run the Program**

Once you have successfully compiled your modified program, ensure that your completed program runs successfully.

STEP 8   **Verify Your Output**

When you enter the above information, the **Console** window should show your program output. Verify that the output is numerically accurate and that the required output information is displayed. Use separate program runs to compare the output when different inputs are used.

PROJECT   Arrays and Methods with Java - Searching and Sorting
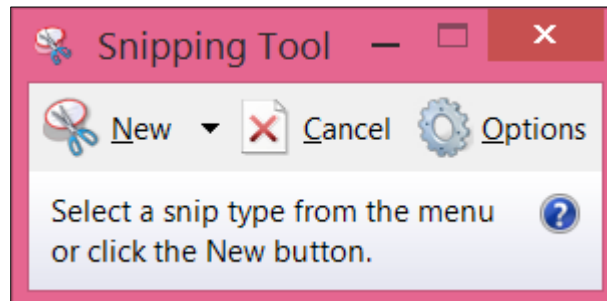
STEP 9   **Submit Your Project**

Once you have determined that your modified program is satisfying this project's requirements, complete the submission process as follows:

Open MS Word and type a heading for a new document that includes your full name, course number, lab number and date.

Within the document paste a snapshot of your program code. Label your snapshot with a reasonable description.

After the snapshot, paste the output that appears in your **Console** screen.

Note - you can use the **Windows Snipping Tool**, which is part of the Windows Accessories Group, to easily capture your **Console** window**.**

Your score for this project will generally be based upon the following factors: documentation, output correctness, content, organization, style and creativity. Submit your Word document to the appropriate course Submittal Box.

STEP 10   **Questions and Answers Concerning this Computer Project**

Answer the following questions in your own words.

Open MS Word and, within your lab submittal document, place your responses to each of these questions. Submit your completed MS Word document for credit.

(1)   Why do programmers often refer to arrays as underlined subscripted variables?

(2)   What is meant by the term underlined parallel arrays?

(3)   The statement below was incorporated in the Linear Search method that was given with the starter code for this application. What was the purpose of this variable declaration? Why do programmers often use the word `flag` for this purpose?

```
int flag = 0;
```

(4)   Why is a Linear Search referred to as " linear "?

(5)   Why is a Bubble sort referred to as a " bubble " type sort?