INTPROG JAVA Practical Worksheet 8

WEBLOG-ANALYZER PROJECT

1. Download and explore the weblog-analyzer project discussed in the lecture. Create a *LogAnalyzer* object and call its *analyzeHourlyData* method. Then call *printHourlyCounts* to discover which is the busiest time of the day.

2. Make sure you understand the for loop used in *printHourlyCounts*. What happens if you replace the condition “hour < hourCounts.length” with the (incorrect) “hour <= hourCounts.length?

3. Replace the for loop in *printHourlyCounts* with an equivalent while loop. Check that the method gives the same results as before.

4. Complete the *numberOfAccesses* method, below, to count the total number of accesses recorded in the log file

   ```
   public int numberOfAccesses()
   {
           int total = 0;
           //code here to add the value of each element
           // of hourCounts to total
           //
           return total;
   }
   ```

5. Modify the LogAnalyzer class so it has a constructor that can take the name of the log file to be analyzed. Have this constructor pass the file name to the constructor of the LogfileReader class. Use the LogfileCreator class to create your own file of random log entries, and analyze. Check that *numberOfAccesses* gives the correct result.

6. Add a method *busiestHour* to LogAnalyzer that returns the busiest hour. You need to look through the *hourCounts* array to find the element with the biggest count. (Hint: you need to look through the entire array).

7. Add a method *quietestHour* to LogAnalyser that returns the number of the least busy hour.

8. Add a method to LogAnalyzer that finds which two hour period is the busiest. Return the value of the first hour in the period.

The Fibonacci Series

The Fibonacci series is a series of integers which begins with:
0 1 1 2 3 5 8 13 21 34……..

The value of fib[n] is defined as follows:

fib[0] = 0 and fib[1] = 1

and for all other integer values of i (i >= 1)

fib(i) = fib[i-1] + fib[i-2]

a. Write a class called Fib20 that stores the first 20 fibonacci numbers in a 20 element int array. The constructor should initialise the array. Write the following methods:

   i. fibtotal – returns the total sum of the first 20 fibanacci numbers
   ii. fibaverage – returns the average of the first 20 fibanacci numbers

b. Create a new class FibN which stores the first ‘n’ Fibonacci numbers in an ‘n’ element int array (where n <= 50). The constructor takes a parameter ‘n’ and creates an array of the appropriate size and initialises it. You should write modified versions of fibtotal and fibaverage methods that work for ‘n’ fibinacci numbers.