Module 1

WMADP 201 Java Programming - MADP 201

Activity 2

Due: 10:30pm, Thursday, June 20th, 2019

Requirements

- Please use meaningful name for your variables and functions
- Try to reuse your solutions as much as possible.
- For each of the following problem you need to
  - Define a class
  - Define required methods (instance or static or combinations)
  - For all test cases you have already written for your algorithm, write a function call inside the main method of the ApplicationDriver class
- Define an ApplicationDrievr class with a main method inside of it.
- Write the test cases for each class in the main method of the ApplicationDriver class.
- There is going to be only one ApplicationDriver class and one main method.
-

Problem0

- Implement a Java class called Problem0 with at least one instance method which receives an ArrayList of integer as its input parameter. The array might have repeated numbers. The method return the number that is repeated the most. Make this assumption that there is only one number which is repeated the most.

Problem1

- Implement a Java class called Problem1 with at least one static method which receives an ArrayList of integer as its input parameter. The array might have repeated numbers. The method will find all numbers that are repeated more than once and print them in the following format:
- Example:

```
4 : 2
10: 4
123: 2
```

Problem2

- Write a Java class with at least one static method which receives a list of integer which may contains repeated numbers. The method will removes the repeated numbers and keeps the distinct number. The method should return the list of distinct numbers.
- The program should print the original list as well as the list of distinct numbers.

Problem3

- Write a Java class with an instance method which receives a list of integers. From each number in the list has been repeated exactly 2 times in the list except one number that is repeated only once. The method should return the number that is repeated only once.

Problem4

- Each student is represented by the following properties:
  - firstName
  - lastName
  - address
  - Year of bird
  - Average

Write a Java class with a static method which receives a list of students as its input parameter and return the student who has the highest average. (Remember: You can use a dictionary to represent a student. Do not define a class for the student at this time)

Problem5

- Write a Java class with an instance method which has no input parameter but return a dictionary. The method let's the user enter names as long as the user enter 0. The user can enter a same name over and over. At the end the method will print and return a dictionary which shows all the name the user has entered and how many times each name is repeated.

Problem6

- Write a Java class with an instance method which has no input parameter. The method ask the user to enter numbers. The user can enter repeated numbers but if the user entered a number which was already entered, the method will provide an error message to the user and ask the user to enter another one. The user can enter numbers as long as it has not entered 0. Once a 0 is entered, the method return the sum of all distinct numbers the user had entered.

Problem7

- Define and implement a Java class with a static method which does linear search. This function receives two input parameters, one is a list of integers and the other one is a number to search for. The method returns -1 if the number (the second parameter of the function) does not exist in the list or return the index of the number if the number exists in the list.
    - If there are more than one occurrence of the number, the function will return the index of the first occurrence

Problem8

- Design and implement a Java class with a static method which receives two input parameters 1) a list of integer numbers and 2) a number. The method will find any occurrence of the given input number in the list and remove the number from the list and finally will return the new list.

Problem9

- Implement a Java program with an instance method which receives a 2D list which represent the following grid:

■							■	■	
		■							
	■	■	■	■	■				
					■			■	
					■			■	
	■	■	■	■	■			■	
					■				
			■	■	■	■			

- The dark cell means they are empty.
- Each cell is associated a number (called weight) based on its row's and column's number. The number is calculated using this formula: (i+j)*3-10. For instance the cell[4][5] = (4+5)*3-10=27-10=17
- The method should find the cell whose total sum of it's neighbour weight is maximum and return a tuple which contains the row and column of the cell. The numbers of a cell is

defined as all cells that have an edge in common with the cell. For instance the neighbours of the green cell is all the red cells: