Java4 Assignment
Aditya Kumar

Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.

1. Write a method that takes a string and returns the number of unique characters in the string.
2. Write a method that takes a string and print the number of occurrence of each character characters in the string.
3. Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Double Age; Double Salary; String Name
4. Write a program to sort the Student objects based on Score , if the score are same then sort on First Name . Class Student{ String Name; Double Score; Double Age
5. Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.
6. Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. (Expected complexity O(1))
7. Write a program to format date as example "21-March-2016"
8. Write a program to display times in different country format.

```
//
import java.util.Arrays;
import java.util.List;
import java.util.Iterator;
class SumOfList{
    public static void main(String[] args) {
        List<Double> list = Arrays.asList(1.0, 2.5, 3.0, 4.5, 5.0, 6.0);
        // Iterator to traverse the list
        Iterator iterator = list.iterator();
        System.out.println("List elements : ");
        while (iterator.hasNext()) {
            System.out.print(iterator.next() + " ");
            System.out.println();
        }
        System.out.print(sumAll(list));
/*
        int total = numbers.stream().mapToInt(i -> i.intValue()).sum();
        System.out.print(total);
        total = numbers.stream().mapToInt(value -> value).sum();
        System.out.print(total);
        total = numbers.stream().mapToInt(Integer::intValue).sum();
        System.out.print(total);
*/
    }
    public static int sumAll(List<Double> numbers) {
        int total = 0;
        for (double number : numbers) {
            total += number;
        }
        return total;
    }
}
```

```
//
import java.util.ArrayList;
class GetUniqueCount {
    public static int getUniqeCount(String arg) {
        ArrayList<Character> unique = new ArrayList<Character>();
        for (int i = 0; i < arg.length(); i++)
            if (!unique.contains(arg.charAt(i)))
                unique.add(arg.charAt(i));
        return unique.size();
    }
}
class GetUniqueCount2{
    public static void main(String[] arg){
        System.out.println(GetUniqueCount.getUniqeCount("adiibtyaa"));
    }
}
```

```
//

import java.util.HashMap;
class OccuranceOfCharacter
{
    static void characterCount(String inputString)
    {
        //Creating a HashMap containing char as a key and occurrences as  a value
        HashMap<Character, Integer> charCountMap = new HashMap<Character, Integer>();
        //Converting given string to char array
        char[] strArray = inputString.toCharArray();
        //checking each char of strArray
        for (char c : strArray)
        {
            if(charCountMap.containsKey(c))
            {
                //If char is present in charCountMap, incrementing it's count by 1
                charCountMap.put(c, charCountMap.get(c)+1);
            }
            else
            {
                //If char is not present in charCountMap,
                //putting this char to charCountMap with 1 as it's value
                charCountMap.put(c, 1);
            }
        }
        //Printing the charCountMap
        System.out.println(charCountMap);
    }
    public static void main(String[] args)
    {
        characterCount("hello java");
        characterCount("All Is Well");
        characterCount("we will miss you");
    }
}
```

```java
//

import java.util.Collections;
import java.util.ArrayList;
import java.util.Comparator;
   class EmployeeBean {
    private int eid;
    private String ename;
    private Integer salary;
    public EmployeeBean(int eid, String ename, Integer salary) {
        super();
        this.eid = eid;
        this.ename = ename;
        this.salary = salary;
    }
    public int getEid() {
        return eid;
    }
    public void setEid(int eid) {
        this.eid = eid;
    }
    public String getEname() {
        return ename;
    }
    public void setEname(String ename) {
        this.ename = ename;
    }
    public Integer getSalary() {
        return salary;
    }
    public void setSalary(Integer salary) {
        this.salary = salary;
    }
    @Override
    public String toString() {
        return "EmployeeBean [eid=" + eid + ", ename=" + ename + ", salary="
                 + salary + "]";
    }
}
class MyComparator implements Comparator<EmployeeBean>{
    @Override
    public int compare(EmployeeBean arg0, EmployeeBean arg1) {
        if (arg0.getSalary()==null && arg1.getSalary()==null) {
            return 0;
        }
        else if (arg0.getSalary()==null) {
            return -1;
        }
        else if (arg1.getSalary()==null) {
            return 1;
        }
        else if (arg0.getSalary()<arg1.getSalary()) {
            return 1;
        }
        else if (arg1.getSalary()<arg0.getSalary()) {
            return -1;
        }
        else
            return 0;
    }
}
public class SortEmployeeByComparator{
    public static void main(String[] args) {
        ArrayList<EmployeeBean> empList=new ArrayList<EmployeeBean>();
        Integer i=null;
        empList.add(new EmployeeBean(1, "kaarthik", 5000));
        empList.add(new EmployeeBean(2, "kaarthik2", 6000));
        empList.add(new EmployeeBean(3, "kaarthik3", 7000));
        empList.add(new EmployeeBean(4, "kaarthik4", 2000));
        empList.add(new EmployeeBean(5, "kaarthik5", 2000));
        System.out.println("Before sorting");
        System.out.println(empList);
        System.out.println("After sorting");
        Collections.sort(empList,new MyComparator());
        System.out.println(empList);
    }
}
```

//

```
import java.util.*;
class sortmapKey {
    // This map stores unsorted values
    static Map<String, Integer> map = new HashMap<>();
    // Function to sort map by Key
    public static void sortbykey()
    {
        // TreeMap to store values of HashMap
        TreeMap<String, Integer> sorted = new TreeMap<>(map);
        // Display the TreeMap which is naturally sorted
        for (Map.Entry<String, Integer> entry : sorted.entrySet())
            System.out.println("Key = " + entry.getKey() +
                        ", Value = " + entry.getValue());
    }
    // Driver Code
    public static void main(String args[])
    {
        // putting values in the Map
        map.put("aditya", 80);
        map.put("Abhishek", 90);
        map.put("amar", 80);
        map.put("deepika", 75);
        map.put("rahul", 40);
        // Calling the function to sortbyKey
        sortbykey();
    }
}
```

//

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
public class SortElementByFrequency6{
    private static void sortByFrequency(int[] arr) {
        Map<Integer, Integer> frequencyMap = createFrequencyMap(arr);
        List<Entry<Integer, Integer>> entryList = sortByValue(frequencyMap);
        putSortedElementsBackInArray(arr, entryList);
    }
```

```java
    private static Map<Integer, Integer> createFrequencyMap(int[] arr) {
        // Use LinkedHashMap because it maintains insertion order of elements.
        Map<Integer, Integer> frequencyMap = new LinkedHashMap<>();
        for (int i = 0; i < arr.length; i++) {
            int key = arr[i];
            if (frequencyMap.containsKey(key)) {
                frequencyMap.put(key, frequencyMap.get(key) + 1);
            } else {
                frequencyMap.put(key, 1);
            }
        }
        return frequencyMap;
    }
    private static List<Entry<Integer, Integer>> sortByValue(
            Map<Integer, Integer> frequencyMap) {
        // List containing elements of map's entry set.
        List<Entry<Integer, Integer>> entryList = new ArrayList<Entry<Integer,
Integer>>(
                frequencyMap.entrySet());
        // Sort the list.
        Collections.sort(entryList,
                new Comparator<Map.Entry<Integer, Integer>>() {
                    @Override
                    public int compare(Entry<Integer, Integer> o1,
                                        Entry<Integer, Integer> o2) {
                        return o2.getValue().compareTo(o1.getValue());
                    }
                });
        return entryList;
    }
    private static void putSortedElementsBackInArray(int[] arr,
                                                       List<Entry<Integer, Integer>>
list) {
        int index = 0;
        // Arrange array elements in sorted list of entry set of frequency map.
        for (Map.Entry<Integer, Integer> entry : list) {
            for (int i = 0; i < entry.getValue(); i++) {
                arr[index++] = entry.getKey();
            }
        }
    }
    private static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }
    public static void main(String[] args) {
        int[] arr = { 2, 5, 3, 8, 7, 2, 5, 2, 3, 3, 5, 3 };
        System.out.println("Input array before sorting elements by frequency.");
        printArray(arr);
        sortByFrequency(arr);
        System.out.println();
        System.out.println();
        System.out.println("Array after sorting elements by frequency.");
        printArray(arr);
    }
}
```

```java
//
public class DynamicStack7{
    private int stackSize;
    private int[] stackArr;
    private int top;
    public DynamicStack7(int size) {
        this.stackSize = size;
        this.stackArr = new int[stackSize];
        this.top = -1;
    }
    public void push(int entry){
        if(this.isStackFull()){
            System.out.println(("Stack is full. Increasing the capacity."));
            this.increaseStackCapacity();
        }
        System.out.println("Adding: "+entry);
        this.stackArr[++top] = entry;
    }
    public int pop() throws Exception {
        if(this.isStackEmpty()){
            throw new Exception("Stack is empty. Can not remove element.");
        }
        int entry = this.stackArr[top--];
        System.out.println("Removed entry: "+entry);
        return entry;
    }
    public long peek() {
        return stackArr[top];
    }
    private void increaseStackCapacity(){
        int[] newStack = new int[this.stackSize*2];
        for(int i=0;i<stackSize;i++){
            newStack[i] = this.stackArr[i];
        }
        this.stackArr = newStack;
        this.stackSize = this.stackSize*2;
    }
    public boolean isStackEmpty() {
        return (top == -1);
    }
    public boolean isStackFull() {
        return (top == stackSize - 1);
    }
    public static void main(String[] args) {
        DynamicStack7 stack = new DynamicStack7(2);
        for(int i=1;i<10;i++){
            stack.push(i);
        }
        for(int i=1;i<4;i++){
            try {
                stack.pop();
            } catch (Exception e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}
```

//

```java
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
public class CalenderPractice{
    public static void main(String[] args) {
        SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
        try {
            Date date = formatter.parse("20/03/2016");
            System.out.println("Date is: "+date);
        } catch (ParseException e) {e.printStackTrace();}
    }
}
```

//

```java
import java.text.DateFormat;
import java.util.*;
public class DifferentCountryFormat{
    public static void main(String[] args) throws Exception {
        Date d1 = new Date();
        System.out.println("today is "+ d1.toString());
        Locale locItalian = new Locale("it","ch");
        DateFormat df = DateFormat.getDateInstance (DateFormat.FULL, locItalian);
        System.out.println("today is in Italian Language  in Switzerland Format : "+
df.format(d1));
    }
}
```