**Scenario:**

A non-profit organization “Green Protect” is involved in conducting programs related to safeguarding nature and provide aid to organic cultivation and cattle breading and other nature protective missions. This being an organization having a lot number of volunteers from different parts of life and different parts of world it getting difficult to maintain and track their volunteer contacts and activities.

Due to a vide spread organization and having a low budget to spend on internet of techno centric distributed application, the organization requested its volunteer to send the daily activity report in the form of a csv text file called “ActivityReport.txt”. Each line of the file represents an activity record and follows the structure. Each field value of the record is separated by “,”. Sample file is attached below for your reference.



**Task**

You have been assigned with a high priority task of implementing **THREE** functions to the existing **Activity Management System.** The below attached Java file (ActivityManager.java) has following classes definitions. Use this Java file to implement the functions.

1. ActivityManager – Activity Manager that tracks the activities.
2. ActivityDetailsVO – Object representation of an activity record.
3. ActivityReportKeyVO – Object representation of an activity report key for grouping activities.
4. ActivityParseException – User defined exception representing abnormalities during activity processing



***Points to remember***

1. *You can use STS/Eclipse IDE with JDK1.8 to develop the code for the implementation. Ensure that you follow best Java coding practice while coding.*
2. *Do not modify any attribute names and method signatures provided as part of the given Java file.*
3. *Code the implementation in the respective methods of the given class ONLY.*
4. *Do not hard code the return values of methods, if found your code will not be considered for processing.*
5. *Use Java SE 8.0 API only for the function implementation.*
6. *Upload/Submit the updated ActivityManager.java file for evaluation.*

Following are the THREE functions to be implemented.

1. **Function 1 – Loading Value objects with activity details**

***Description***

**Read the file “Activity**Report.txt” from a file path (record by record) and populate the field values in value objects. One value object for one record. Store all the value objects in a List and return the list. Handle all possible exceptions with respective handlers.

***Implementation***

Class: **ActivityManager**

Method: **public List<ActivityDetailsVO> loadAllActivityDetails(String filePath) throws ActivityParseException**

You can write additional utility methods wherever needed.

***Input***

Absolute path of the file “**Activity**Report.txt”.

***Output:***

List of value objects representing activities.

***Validations***

While reading the activity records from the file ensure that the following validations are performed, if violated raise InvalidActivityUncheckedException and skip that record.

Activity Id – Should be a non-zero positive number

Title – Non-empty. Should be a String of 3 to 20 characters

Expected Conduction Date – Non-empty and should be a past or present date.

Actual Conduction Date – Non-empty and should be a date later to expectedConductionDate

***Constraints:***

Use java.time.format.DateTimeFormatter while formatting the Date entries.

Skip the first line in the input text file “ActivityReport.txt” as it contains headings.

1. **Function 2 – Retrieving all activities related to a given Location.**

***Description***

**Read the file “ActivityReport.**txt” from a file path (record by record) and return a Map with the keys as Location and the values as Set of Activities having the same Location as the key. The activities in the set need to be ordered by Actual Conduction Date (in descending order). The map entries should be in the ascending order of the key names.

***Implementation***

Class: **ActivityManager**

Method: **public Map<String, Set<ActivityDetailsVO>> getSearchedActivityDetails(String filePath, String searchText) throws ActivityParseException**

You can write additional utility methods wherever needed.

***Input***

Absolute path of the file “ActivityReport.txt” and the Location that has to be searched

***Output***:

Map with the keys as Location and the values as Set of ActivityDetailsVO.

*Validations*

Throw ActivityParseException if anything goes wrong during the output Map construction.

*Constraints*:

Use loadAllActivityDetails() method to read the issue records from the input file.

Use ONLY java.util.stream package and lambda expression while searching for the activity and creating the result Map.

*Sample Input:*

**ActivityReport.txt,** **Banglore**

ActivityId,Title,Location,Budget,ExpectedConductionDate,ActualConductionDate,PrecedingOfficer,Participants

1,CleanAndGreen,Visakhapatnam,22000,10-11-2020,28-11-2020,Vamsy,Sarat;Suma;Vani

2,CleanAndGreen,Banglore,40000,11-11-2020,29-11-2020,Vamsy,Mary;Dhamu;Praveen

3,CleanAndGreen,Vijayawada,30000,12-11-2020,30-11-2020,Vamsy,Rahul;Kala;Sreenu

4,CleanAndGreen,Pune,35000,13-11-2020,01-12-2020,Vamsy,Sivaji;Mahreen;Subha

5,OrganicCultivation,Pune,36000,14-11-2020,02-12-2020,Vamsy,Sivaji;Mahreen;Subha

6,OrganicCultivation,Vijayawada,29000,10-11-2020,30-11-2020,Srinu,Rahul;Kala;Sreenu

7,PollutionControl,Banglore,20000,11-11-2020,01-12-2020,Srinu,Mary;Dhamu;Praveen

8,PollutionControl,Visakhapatnam,24500,12-11-2020,02-12-2020,Srinu,Sarat;Suma;Vani

9,PaperLessDining,Banglore,35000,13-11-2020,28-11-2020,Srinu,Mary;Dhamu;Praveen

10,PollutionControl,Pune,36000,14-11-2020,29-11-2020,Srinu,Sivaji;Mahreen;Subha

*Sample Output:*

**Map with String as Key and value as list of Value Objects representing issues.**

|  |  |
| --- | --- |
| **Key** | **Value** |
| Banglore | 2,CleanAndGreen,Banglore,40000,11-11-2020,29-11-2020,Vamsy,Mary;Dhamu;Praveen |
| 7,PollutionControl,Banglore,20000,11-11-2020,01-12-2020,Srinu,Mary;Dhamu;Praveen |
| 9,PaperLessDining,Banglore,35000,13-11-2020,28-11-2020,Srinu,Mary;Dhamu;Praveen |

1. **Function 3 – Retrieving all activities grouped by activity title and preceding officer.**

***Description***

**Read the file “ActivityReport.**txt” from a file path (record by record) and return a Map with the keys as ActivityReportKey (activity title, preceding officer) and the values as List of Activities having the same location and preceding officer as the key.

***Implementation***

Class: **ActivityManager**

Method: **public Map<ActivityReportKey, List<ActivityDetailsVO>> getGroupedActivityDetails(String filePath) throws ActivityParseException**

You can write additional utility methods wherever needed.

***Input***

Absolute path of the file “ActivityReport.txt”.

***Output***:

Map with the keys as ActivityReportKey and the values as Set of ActivityDetailsVO.

*Validations*

Throw ActivityParseException if anything goes wrong during the output Map construction.

*Constraints*:

Use loadAllActivityDetails() method to read the issue records from the input file.

Use ONLY java.util.stream package and lambda expression while grouping for the activity and creating the result Map.

*Sample Input:*

**ActivityReport.txt**

ActivityId,Title,Location,Budget,ExpectedConductionDate,ActualConductionDate,PrecedingOfficer,Participants

1,CleanAndGreen,Visakhapatnam,22000,10-11-2020,28-11-2020,Vamsy,Sarat;Suma;Vani

2,CleanAndGreen,Banglore,40000,11-11-2020,29-11-2020,Vamsy,Mary;Dhamu;Praveen

3,CleanAndGreen,Vijayawada,30000,12-11-2020,30-11-2020,Vamsy,Rahul;Kala;Sreenu

4,CleanAndGreen,Pune,35000,13-11-2020,01-12-2020,Vamsy,Sivaji;Mahreen;Subha

5,OrganicCultivation,Pune,36000,14-11-2020,02-12-2020,Vamsy,Sivaji;Mahreen;Subha

6,OrganicCultivation,Vijayawada,29000,10-11-2020,30-11-2020,Srinu,Rahul;Kala;Sreenu

7,PollutionControl,Banglore,20000,11-11-2020,01-12-2020,Srinu,Mary;Dhamu;Praveen

8,PollutionControl,Visakhapatnam,24500,12-11-2020,02-12-2020,Srinu,Sarat;Suma;Vani

9,PaperLessDining,Banglore,35000,13-11-2020,28-11-2020,Srinu,Mary;Dhamu;Praveen

10,PollutionControl,Pune,36000,14-11-2020,29-11-2020,Srinu,Sivaji;Mahreen;Subha

*Sample Output:*

**Map with String as Key and value as list of Value Objects representing issues.**

|  |  |
| --- | --- |
| **Key** | **Value** |
| CleanAndGreen,Vamsy | 1,CleanAndGreen,Visakhapatnam,22000,10-11-2020,28-11-2020,Vamsy,Sarat;Suma;Vani  2,CleanAndGreen,Banglore,40000,11-11-2020,29-11-2020,Vamsy,Mary;Dhamu;Praveen  3,CleanAndGreen,Vijayawada,30000,12-11-2020,30-11-2020,Vamsy,Rahul;Kala;Sreenu  4,CleanAndGreen,Pune,35000,13-11-2020,01-12-2020,Vamsy,Sivaji;Mahreen;Subha |
| OrganicCultivation,Vamsy | 5,OrganicCultivation,Pune,36000,14-11-2020,02-12-2020,Vamsy,Sivaji;Mahreen;Subha |
| OrganicCultivation,Srinu | 6,OrganicCultivation,Vijayawada,29000,10-11-2020,30-11-2020,Srinu,Rahul;Kala;Sreenu |
| PollutionControl,Srinud | 7,PollutionControl,Banglore,20000,11-11-2020,01-12-2020,Srinu,Mary;Dhamu;Praveen  8,PollutionControl,Visakhapatnam,24500,12-11-2020,02-12-2020,Srinu,Sarat;Suma;Vani  10,PollutionControl,Pune,36000,14-11-2020,29-11-2020,Srinu,Sivaji;Mahreen;Subha |
| PaperLessDining,Srinu | 9,PaperLessDining,Banglore,35000,13-11-2020,28-11-2020,Srinu,Mary;Dhamu;Praveen |