**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 is getting difficult to maintain and track their volunteer contacts and activities.

**Task**

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

1. BudgetManager– Budget Manager that tracks the budget spent on each activity.
2. ActivityDetailsVO – Object representation of an activity record.
3. Aggregate– Enum representation three aggregate operations SUM, MIN, MAX.
4. IllegalFieldNameException– User defined exception to be raised if a non-relevant field name supplied as argument to expecting functions.
5. ActivityDetailsRepo – Is a repository of Activity records and supplies a list of hardcoded ActivityDetailsVO objects each of which represents one activity record.



***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 – Field Wise Budget Aggregate**

***Description***

**Retrieve the activity details data from the ActivtyDetailsRepo.getActivities() function**. Prepare a report of total budget/Minimum Budget/Maximum Budget spent on activities grouped by title/location/precidingOfficer. Throw IllegalFieldNameException if the given field is not one amongst title/location/precidingOfficer.

***Implementation***

Class: **BudgetManager**

Method: **public Map<String, Double> getFieldWiseBudget(String field, Aggregate aggregate) throws IllegalFieldNameException**

You can write additional utility methods wherever needed.

***Input***

***Field : “location” or “title” or “precedingOfficer”,***

***Aggregate : A value from aggregate enum (sum/min/max)***

***Output:***

Map with the values of the given field as keys and their respective budget aggregates as values.

***Constraints:***

Use only streams api.

*Sample Input:*

“title”,Aggregate.SUM

*Sample Output:*

*A map having below key, value entries.*

"CleanAndGreen",127000.0

"OrganicCultivation",65000.0

"PollutionControl",80500.0

"PaperLessDining",35000.0

1. **Function 2 – Retrieve that activities having the highest budget spent.**

***Description***

**Retrieve the activity details data from the ActivtyDetailsRepo.getActivities() function**. Using appropriate Steams Operation find out the activity that has the highest budget spent.

***Implementation***

Class: **BudgetManager**

Method: **public ActivityDetailsVO getCostliestActivity() throws IllegalFieldNameException**

You can write additional utility methods wherever needed.

***Input***

No parameters.

***Output***:

ActivityDetailsVO Object representing the activity with the highest budget.

*Constraints*:

UseActivityDetailsRepo.getActivites() to retrieve activities list.

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

*Output:*

ActivityDetailsVO [activityId=2, title=CleanAndGreen, location=Banglore, budget=40000.0, activityDate=2020-11-11, precedingOfficer=Vamsy]

1. **Function 3 – Retrieving Total Budgest Spent Month Wise.**

***Description***

**Retrieve the activity details data from the ActivtyDetailsRepo.getActivities() function**. Compute the total budget spent each month where month represented by MMM-yyyy format and return a map having the month in MMM-yyyy format as key and total budget spent as value.

***Implementation***

Class: **BudgetManager**

Method: **public Map<String, Double> getMonthWiseBudget()**

You can write additional utility methods wherever needed.

***Input***

No arguments

***Output***:

Map with the keys as Month in MMM-yyyy format and the values as total budget spent in that month.

*Constraints*:

UseActivityDetailsRepo.getActivites() to retrieve activities list.

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

*Output:*

**Map as below**

|  |  |
| --- | --- |
| **Key** | **Value** |
| Nov-2020 | 261000.0 |
| Dec-2020 | 22000.0 |
| Apr-2020 | 24500.0 |