**DESCRIPTION**

Your task here is to implement **Java** code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider **default visibility** of classes, data fields, and methods unless mentioned.

**Specifications**

**class** **definitions**:

**class** **Mobile**:

data member:

﻿﻿ HashMap<String, ArrayList<String>> mobileList = **new** Hashmap<>()

﻿ method definition:

addMobile(String company, String model)

**return** type: String

visibility: **public**

getModels(String company)

**return** type: ArrayList<String>

visibility: **public**

buyMobile(String company, String model)

**return** type: String

visibility: **public**

**Task**

**Class** **Mobile**

**-define the object of HashMap<String, ArrayList<String>>** with variable name **mobileList**.

* The **String**defines the **name of the company**and the **Arraylist**will have list of models.

**Implement the below methods for this class:**

**-String addMobile(String company, String model):**

* Write a code to add **a company** with its **model**.
* If the company does not exists then create it with a new String list and add the model.
* Update the String list with the new model if the company already exists
* return "**model successfully added**" after performing the above operations

**-ArrayList<String> getModel(String company):**

* Write a code to get the Model list.
* return null if the given company doesn't exist or doesn't have any model, else return the String list of all the models.

**-String buyMobile(String company, String model):**

* Write a code to buy a mobile.
* Remove the mobile model from the list according to the model purchased. In case there are two same models then remove one and return the message "mobile sold successfully
* Return a message "**item not available" i**f the mobile or model is not present in the list

**Sample Input**

Mobile obj = **new** Mobile();

obj.addMobile("Oppo", "K3");

obj.getModels("Oppo");

obj.buyMobile("Oppo", "K3");

**Sample Output**

model successfully added

[K3]

mobile sold successfully

class Mobile{

HashMap<String, ArrayList<String>> mobileList = new HashMap<>();

public String addMobile(String company,String model){

if(!(mobileList.containsKey(company))){

mobileList.put(company , new ArrayList<String>());

mobileList.get(company).add(model);

}

else{

mobileList.get(company).add(model);

}

return "model successfully added";

}

public ArrayList<String> getModels(String company){

if(!(mobileList.containsKey(company))){

return null;

}

else{

if(!(mobileList.get(company).isEmpty()))

return mobileList.get(company);

else

return null;

}

}

public String buyMobile(String company, String model){

if (!(mobileList.containsKey(company))){

return "item not available";

}

ArrayList<String> a = new ArrayList<>();

a.addAll(mobileList.get(company));

if (!(a.contains(model))){

return "item not available";

}

mobileList.get(company).remove(model);

return "mobile sold successfully";

}

}

public class Source {

public static void main(String args[] ) throws Exception {

Mobile obj = new Mobile();

System.out.println(obj.addMobile("Oppo","K3"));

System.out.println(obj.getModels("Oppo"));

System.out.println(obj.buyMobile("Oppo","K3"));

}

}

Your task here is to implement **Java** code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider **default visibility** of classes, data fields, and methods unless mentioned.

### **Specifications**

class definitions:

class Company:

data members:

String companyName

String type

int turnover

visibility: private

﻿

﻿ Company(String companyName, String type, int turnover): constructor **with** **public** visibility

**Define** getter **and** setter **with** **public** visibility﻿

**class** Record:

**data** members :

**List**<Company> companies

visibility : **public**

﻿

method definition:

addCompanyCompany company):

**return** : **String**

visibility **public**

filterData(**String** **query**):

**return** : **String**

visibility : **public**

**class Company**

- define data members according to the above specifications

-define a constructor and getters setters according to the above specifications

**class Record**

- define data members according to the above specifications

-Implement the below methods for this class:

**-String addCompany(Company company):**

* ﻿Write a code to add a given company object to the company ArrayList.
* Add the company object to the companies list and return **"Added".**

**-String filterData(String query):**

* ﻿Write a code that filters the data according to the given query and returns valid/filtered data.
* A query is of a string data type with 3 entities. Consider the given format - "type == "Mycompany"".
* In the above-defined query 1st entity "type" defines the attribute of the company, 2nd entity represents the operator and 3rd entity represents value.
* Consider the following condition to implement the method -

1. return "Invalid query" if there are less than 3 entities in the value.
2. return "Invalid operator" if the operator is other than "==", ">=" and "<=".
3. If 1st entity is a "type" and the operator is other than "==" then return "Invalid operator".
4. If the 1st entity is other than "type" then return "Invalid"
5. If 1st entity is a "type" and the operator is equal to "==" then call byType method and return the string send by it.

**Sample Input**

 Company c1 = **new** Company("Doselect","IT",300);

  Record record = **new** Record();

  record.addCompany(c1);

**Sample Output**

**Added**

**Explanation**

Company object is created c1 with given parameters and added to the companies list.

class Company{

//Define all the variables and methods here

private String companyName;

private String type;

private int turnover;

public Company(String companyName, String type, int turnover){

this.companyName = companyName;

this.type = type;

this.turnover = turnover;

}

public String getCompanyName(){

return this.companyName;

}

public void setCompanyName(String companyName){

this.companyName = companyName;

}

public String getType(){

return this.type;

}

public void setType(String type){

this.type = type;

}

public int getTurnover(){

return this.turnover;

}

public void setTurnover(int turnover){

this.turnover = turnover;

}

}

class Record{

//Define all the variables and methods here

public ArrayList<Company> companies = new ArrayList<Company>();

public String addCompany(Company company){

companies.add(company);

return "Added";

}

public String filterData(String query){

String[] arr = query.split(" ");

{

if(arr.length<3)

return "Invalid query";

else if(!(arr[1].equals("<=")||arr[1].equals(">=")||arr[1].equals("==")))

return "Invalid operator";

}

{

if(arr[0].equals("type")){

if(arr[1].equals("=="))

return byType(arr[2]);

else

return "Invalid operator";

}

else if(arr[0].equals("turnover"))

return byTurnOver(arr[1],arr[2]);

else

return "Invalid entity";

}

}

public String byType(String value){

String answer = "";

for(int i=0;i<companies.size();++i){

Company current = companies.get(i);

if(current.getType().equals(value)){

answer+=current.getCompanyName();

answer+=",";

}

}

return answer.substring(0,answer.length()-1);

}

public String byTurnOver(String operator, String value){

String answer = "";

for(int i=0;i<companies.size();i++){

Company current = companies.get(i);

if(operator.equals("==")){

if(current.getTurnover() == Integer.parseInt(value)){

answer+=current.getCompanyName();

answer+=",";

}

}

else if(operator.equals(">=")){

if(current.getTurnover() >= Integer.parseInt(value)){

answer+=current.getCompanyName();

answer+=",";

}

}

else{

if(current.getTurnover() <= Integer.parseInt(value)){

answer+=current.getCompanyName();

answer+=",";

}

}

}

return answer.substring(0,answer.length()-1);

}

}

// Class name should be "Source",

// otherwise solution won't be accepted

public class Source {

public static void main(String args[] ) throws Exception {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

}

}

**Problem Statement**

Complete the class**Population**using the Specifications given below. Consider default visibility of classes, data fields, and methods unless mentioned otherwise.

**Specifications**

**class** **definitions**:

﻿**class** **Population**:

data member:

﻿HashMap<String,Integer> dataList

method definitions:

maxPopulation():

﻿**return** type: String

minPopulation():

**return** type: String

totalPopulation():

**return** type: **long**

**Task**

Class **Population**

**-**define the **HashMap<String, String>**variable **dataList.**

* Keys **->countryName** .
* values -> **population.**

Implement the below methods for this class:

-**String** **maxPopulation():**

* Write the code to get the country name with the maximum population.
* If there is more than one country having maximum **population** then return any.

Example: dataList = {"India": 20000, "USA":1098, "Uk":20000} then output can be "India" or "UK".

**-String minPopulation():**

* Write the code to get the country name with the minimum population.
* If there is more than one country having a minimum **population** then return any.

Example: dataList = {"Nepal": 1098, "Bhutan":1098, "Uk":20000} then output can be "Nepal" or "Bhutan".

**-long totalPopulation():**

* Write the code to calculate the total population of all the countries.
* **totalPopulation** = P1+P2+. . . +Pn.

Example: dataList = {"Nepal": 1098, "Bhutan":1098, "Uk":20000} then output = 22196.

Population obj = **new** Population();

obj.dataList.put("India",26382898);

obj.dataList.put("USA",82738);

obj.dataList.put("UK",92732);

String max = obj.maxPopulation();

String min = obj.minPopulation();

long total = obj.totalPopulation();

**Sample Output**

max = India

min = USA

total = 26558368

class Population{

HashMap<String, Integer> dataList = new HashMap<>();

String maxPopulation()

{

int max = 0;

String maxCount = "";

for(Map.Entry<String, Integer> entry: dataList.entrySet())

{

if(max<entry.getValue())

{

max = entry.getValue();

maxCount = entry.getKey();

}

}

return maxCount;

}

String minPopulation()

{

String result = "";

// Collection set = ;

int min = dataList.get(this.maxPopulation());

// Object min = Collections.min(dataList.values());

for(Map.Entry<String, Integer> entry: dataList.entrySet())

{

// if(min == entry.getValue())

// {

// result += entry.getKey();

// }

if(min>entry.getValue())

{

min = entry.getValue();

result = entry.getKey();

}

}

return result;

}

long totalPopulation()

{

long totalPop = 0;

for(Map.Entry<String, Integer> entry : dataList.entrySet())

{

totalPop +=entry.getValue();

}

return totalPop;

}

}

public class Source {

public static void main(String args[] ) throws Exception {

Population obj = new Population();

obj.dataList.put("India", 26382898);

obj.dataList.put("USA", 82898);

obj.dataList.put("UK", 92898);

String max = obj.maxPopulation();

String min = obj.minPopulation();

long total = obj.totalPopulation();

System.out.println(max);

System.out.println(min);

System.out.println(total);

}

}