CIS 263 Winter 2022 Project 1

In this project you will write methods in an existing project. The purpose is to practice writing code that uses classes and methods in the Collections framework in Java. The skeleton of the project is provided.

The project is called “Olympic Medalists”. The project reads a file with information of the Olympic medalists of several Olympiads. The information about the individual athletes is stored in an ArrayList of Olympic Medalists. A consolidation process is performed to summarize the information by country and to store that information in a separate ArrayList. The user can execute seven different queries that operate on these two ArrayLists. These queries fall into three categories:

1. Queries about individual medalists:
    a. Who won medals in a particular year.
    b. Who were the medalists from a particular country.
    c. Find medalists by their name.
2. Queries about countries:
    a. How many medals did each country win in a particular year?
    b. Which were the top ten countries with gold medals in a particular year.
    c. Which country won the most bronze medals in a particular year
3. Counts:
    a. How many medalists are in the database, how many women athletes are in the database, how many men athletes are in the database and the sum of countries that won Olympic medals in different years (if a country won Olympic medals in two different years, it will be counted twice).

In the attachments you will find the following source code files:

- Olympic Medalist.java
- OlympicCountryMedals.java
- OlympicMedalistsDatabase.java
- OlymmpicMedalistsGUI.java

Two files with test cases:

- MyOlimpicMedalistTestPhase1.java
- MyOlimpicMedalistTestPhase2.java

And the comma separated values file with the information about the athletes:

- SummerOlympicMedals.csv

You will need to complete two methods in the class OlympicMedalistsDatabase.java

The method genereateCountryTotalMedals()

When the program read the file, every record in the input file contains the information about an individual athlete. The fields are city, year, sport, discipline, event, athlete's name, gender, country code and the type of medal (Gold, Silver or Bronze). Every one of these records is used to create an object of the class OlympicMedalist, which is added to the ArraryList om, which is an ArrayList of OlympicMedalists. Once the information about all the athletes has been read, the method genereateCountryTotalMedals() is invoked to generate the consolidated information by country. The information for every country for every year will be stored in an object of the class OlympicCountryMedals. The OlympicMedalistsDatabase in turn has an ArrayList of objects of the class OlympicCountryMedals called ocm.

The OlympicCountryMedal object has the following attributes:

```
private String city;

private int year;

private String countryCode;

private int goldMedals;

private int silverMedals;

private int bronzeMedals;
```

The first method that you need to write is the method public void generateCountryTotalMedals().

Instantiate a Map where the key is going to be a string and the value will be an array of integers. You will use the array of integers to hold the number of gold medals in entry 0, the number of silver medals in entry 1 and the number of bronze medals in entry 2. Let's call this map totals.

Go over all the entries in the array om of Olympic Medalsts. For every record do the following:

Create a string, let's call it key, as the concatenation of the city, the year, and the country code. Separate those with commas.

Using the string key, check it the Map totals contains the key. Use the method containsKey.

If the Map totals contains the key, this means you have started to collect statistics for this country for this Olympic games already. Retrieve into an array of integers, let's call it medals, the current value of the array stored in totals. Use the method get with the String key as a parameter.

If the medal for this entry in om is a Gold medal, increment by the entry in position 0 in medals.

If the medal for this entry in om is a Silver medal, increment by the entry in position 1 in medals.

If the medal for this entry in om is a Bronze medal, increment by the entry in position 2 in medals.

If totals does not contain the key, this means this is the first time you have found an athlete from this country who has won a medal at these games. Instantiate an array, let's call it again medals of 3 integers. Again:

If the medal for this entry in om is a Gold medal, increment by the entry in position 0 in medals.

If the medal for this entry in om is a Silver medal, increment by the entry in position 1 in medals.

If the medal for this entry in om is a Bronze medal, increment by the entry in position 2 in medals.

Add this new pair (key,totals) into the map totals. Use the method put and pass as parameters the string Key and the array medals.

After you have processed all the entries in the ArrayList of Olympic Medalists, traverse the entire Map using a for statement like this:

for(Map.Entry<String,int[]> entry: totals.entrySet())

Retrieve the key using the method entry.getKey(). Split this string using "," as a separator and this will give you the city, the year and the country code. You will need to convert the year to an integer.

Retrieve the array that counts of medals using entry.getValue(). This will return an array with three entries.

Now you can instantiate an object of the class OlympicCountryMedals, passing as parameters the year, the city, the country code and the entries in the array of totals of medals. Add this object to the ArrayList ocm.

You can check that your method is working correctly by using the methods testReadFileAndCounts(), testsearchMedalistsByCountry() and testfindCountryWithHighestBronzeMedalsByYear() in the Unit test case MyOlimpicMedalistTestPhase2.

The Method public ArrayList < OlympicCountryMedals > topTenCountriesGoldMedals (int year)

The purpose of this method is to find which were the ten countries with the most gold medals at the event of a particular year.

Start by declaring an ArrayList of OlympicCountryMedals, let's call it result.

Go over all the OlympicCountryMedals objects in the ArrayList ocm. If the year for a particular object matches the parameter year, add it to the ArrayList result.

At this point in time the ArrayList result may have more than ten entries.

If the size of result is bigger than 10, use the method Collections.sort(result). This will sort the ArrayList result by the number of gold medals in descending order. The class OlympicCountryMedals implements the interface Comparable and therefore it is feasible to use the static sort method in the class Collections.

Then use the subList method on the object result to extract the first ten objects. The method subList will return an object of the class List. Pass this object to the class List to a constructor for an ArrayList and return this ArrayList.