This is a short exercise in using the stream API. Suppose that the class *Score* is defined as

class ScoreInfo {

String firstName;

String lastName;

int score;

ScoreInfo( String lName, String fName, int s ) {

firstName = fName;

lastName = lName;

score = s;

}

}

and that scoreData is an array of *ScoreInfos* containing information about the scores of students on a test. Use the stream API to do each of the following tasks:

* print the number of students (without using scoreData.length)
* print the average score for all of the students
* print the number of students who got an A (score greater than or equal to 90)
* use the collect() stream operation to create a *List<String>* that contains the names of students whose score was less than 70; the names should be in the form first name followed by last name
* print the names from the *List* that was generated in the previous task
* print out the student's names and scores, ordered last name
* print out the student's names and scores, ordered by score

You can put all of the code in main() routine and include *ScoreInfo* as a static nested class. Do not use **any** for loops or other control structures. Do everything using the stream API. For testing your code, you can use this array:

private static ScoreInfo[] scoreData = new ScoreInfo[] {

new ScoreInfo("Smith","John",70),

new ScoreInfo("Doe","Mary",85),

new ScoreInfo("Page","Alice",82),

new ScoreInfo("Cooper","Jill",97),

new ScoreInfo("Flintstone","Fred",66),

new ScoreInfo("Rubble","Barney",80),

new ScoreInfo("Smith","Judy",48),

new ScoreInfo("Dean","James",90),

new ScoreInfo("Russ","Joe",55),

new ScoreInfo("Wolfe","Bill",73),

new ScoreInfo("Dart","Mary",54),

new ScoreInfo("Rogers","Chris",78),

new ScoreInfo("Toole","Pat",51),

new ScoreInfo("Khan","Omar",93),

new ScoreInfo("Smith","Ann",95)

};

Stream API Exercises : Part II

Q.1) Find the most populated city of each continent

Q.2) Find the number of movies of each director

Q.3) Find the number of genres of each director's movies

Q.4) Find the highest populated capital city

Q.5) Find the highest populated capital city of each continent

Q.6) Sort the countries by number of their cities in descending order

Q.7) Find the list of movies having the genres "Drama" and "Comedy" only

Q.8) Group the movies by the year and list them

Q.9) Sort the countries by their population densities in descending order ignoring zero population countries

Q.10) Find the richest country of each continent with respect to their GNP (Gross National Product) values.

Q.11) Find the minimum, the maximum and the average population of world countries.

Q.12) Find the minimum, the maximum and the average population of each continent.

Q.13) Find the countries with the minimum and the maximum population.

Q.14) Find the countries of each continent with the minimum and the maximum population.

Q.15) Group the countries by continent, and then sort the countries in continent by number of cities in each continent.

Q.16) Find the cities with the minimum and the maximum population in countries.

Q.17) Find the minimum, the maximum, the average, and the standard deviation of GNP values.

Q.18) Find the year where the maximum number of movie is available

1. public class City {

private int id;

private String name;

private int population;

private String countryCode;

//getter & setter

}

2. public class Country {

private String code;

private String name;

private String continent;

private double surfaceArea;

private int population;

private double gnp;

private int capital;

private List<City> cities;

{

cities = new ArrayList<>();

}

//getter & setter

}

3. public class Director {

private int id;

private String name;

private String imdb;

//getter & setter

}

4. public class Genre {

private int id;

private String name;

//getter & setter

}

5. public class Movie {

private int id;

private String title;

private int year;

private String imdb;

private List<Genre> genres;

private List<Director> directors;

{

genres = new ArrayList<>();

directors = new ArrayList<>();

}

}