CSE 216 – Programming Abstractions (Spring 2019)
Programming Assignment # 3

In this assignment, you will create Java and Python programs which use functional programming. The assignment has three parts together comprising 50 points.

The data presented in this assignment is taken from World Total Population dataset[1] made available by The World Bank. This dataset contains a CSV dataset file YearlyPopulation.csv presenting data about population of various countries from the year 1960 to 2017. The first column of the file is country name, second column is country code and rest of the columns show yearly population.

Part I. Java and Python based statistical analysis (20 points for each language implementation)

In this part, you will create a command line-based user interface program that shows a particular statistics requested by a user. The program will be developed in both, Java as well as Python languages using NetBeans and PyCharm IDEs respectively.

Your tasks are the following:

1) **Preprocessing:** This step involves reading YearlyPopulation.csv file and store processed data row-wise using data structure of your choice. You may use Opencsv package (http://opencsv.sourceforge.net/) for reading csv files in Java. A CSV package (https://realpython.com/python-csv/) can be used in Python for reading CSV files.

2) **Create country-wise statistics using Lambda expressions:** Write functions in Java and Python which exclusively use Lambda expressions respectively to calculate the following statistical parameters:
    a. Mean of population between any two years,
    b. Median of population between any two years,
    c. Mode of population between any two years,
    d. Standard Deviation of population between any two years,
    e. Variance of population between any two years, and
    f. Percentage change in population between any two years.

   Each of above functions should take as input a list that contains population data between two years. Basically, you need to provide Lambda expressions-based Java and Python implementation of various functions in MathLib class used in Assignment 1.

   Note: If you do not use Lambda expressions for calculating statistical parameters, you will be awarded zero points for the Part I of the assignment.

3) **User interaction:** Both, for the Java and Python programs, you will present 7 options (including one exit option) corresponding to above statistical parameters to a user. In case the user doesn't

[1] Population, total Data, The World Bank, https://data.worldbank.org/indicator/sp.pop.totl, Last accessed: April 24, 2019.

select Exit, then the program will prompt user to enter country code. In case a proper country code entered, the program will prompt user to select From year followed by To year between which above data is required. In case the correct years are chosen, the selected statistics for those years is shown. Your program will interact in a loop with the user as long as user selects Exit option. In case user enters the input outside allowed values, an error message should be given and user should be prompted again to enter valid input.

4) **Packaging:**

   a. The Java project is named **PopulationDataProcesing**. Use three different packages in your code – corresponding to **preprocessing**, **statistics** and **interaction**. The package naming follows the convention: edu.sunyk.cse216.<packagename>.
   The main program will be in **interaction** package with name **userinterface.java**. The main program will take as command-line input path to the dataset file YearlyPopulation.csv. The program should run in NetBeans when path to the dataset file is provided as program runtime argument.
   Rubric:
   Preprocessing: 5 points
   Statistics lib which uses Lambda expressions: 10 points
   User interaction: 5 points

   b. The Python project is named **PopulationDataProcesing**. Use three different Python scripts in your code – **preprocessing.py**, **statistics.py** and **interaction.py**.
   The main program **interaction.py** will take as command-line input path to the dataset file YearlyPopulation.csv. The interaction.py program will import preprocessing and statistics scripts as required. The program should run from the command line when path to the dataset file is provided as program runtime argument.
   Rubric:
   Preprocessing: 5 points
   Statistics which uses Lambda expressions: 10 points
   User interaction: 5 points

An example of user interaction is shown below:

```
Welcome! Select one of the following statistical parameters:
    1. Mean population
    2. Median population
    3. Mode of population
    4. Standard deviation of population
    5. Variance of population
    6. Percentage change in population
    7. Exit

User selects -> 1

Select country of interest:
```

Country Code

```
  1.   ABW
  2.   AFG
  3.   AGO
  4.   ALB
  5.   AND
  6.   ARB
  7.   ARE
  8.   ARG
  9.   ARM
  10.  ASM
  11.  ATG
  12.  AUS
  13.  AUT
  14.  AZE
  15.  BDI
  16.  BEL
  17.  BEN
  18.  BFA
  19.  BGD
  20.  BGR
  21.  BHR
  22.  BHS
  23.  BIH

  ...

  247.VCT
  248. VEN
  249. VGB
  250. VIR
  251. VNM
  252. VUT
  253. WLD
  254. WSM
  255. XKX
  256. YEM
  257. ZAF
  258. ZMB
  259. ZWE

User selects -> 1

Select From year:

   1.  1960
   2.  1961
   3.  1962
   4.  1963

   ...

   53. 2012
```

```
    54. 2013
    55. 2014
    56. 2015
    57. 2016
    58. 2017

  User selects -> 2

  Select To year:

    1.  1960
    2.  1961
    3.  1962
    4.  1963

    ...

    53. 2012
    54. 2013
    55. 2014
    56. 2015
    57. 2016
    58. 2017

  User selects -> 13

  Output: The mean population of country Aruba from year 1961 to 1973
  is 57831.25.
```

5) **Submission:**
   a. **Java:** Submit your NetBeans project as a Zip file on blackboard.
   b. **Python:** Submit three scripts preprocessing.py, statistics.py and interaction.py.

Part II. MathLib using Java functional interface (10 points)

Work on this part after completing Part I of the assignment. In this part, you will create and use MathLib utilities which implement following Java functional interface:

```
@FunctionalInterface

interface StatsUtil<T,R> {

    R calculateStats(T l);

}
```

Following utils will be used:

a. Mean of population between any two years,
b. Median of population between any two years,
c. Mode of population between any two years,
d. Standard Deviation of population between any two years,

e. Variance of population between any two years, and

f. Percentage change in population between any two years.

E.g. The meancalculator can be created and used as follows:

```
StatsUtil<List<Integer>, Double> meancalculator = l -> l.stream().collect
(Collectors.averagingDouble(x -> x));
```

```
Meancalculator.calculateStats(someList);
```

Create a new Java project **PopulationDataProcesing2** which is basically a copy of Java project **PopulationDataProcesing** created in Part I. The difference is that **PopulationDataProcesing2** will use the Statistics utilities created in Part II. The packaging and user interaction are same as in Part I.

Rubric:

Preprocessing: 2 points
Statistics lib which uses Java functional interface: 6 points
User interaction: 2 points
Note: If you do not use Java functional interface for calculating statistical parameters, you will be awarded zero points for the Part II of the assignment.

Part II Submission: Submit your NetBeans project as a Zip file on blackboard.

Assignment Evaluation: A slot will be announced for evaluating the assignments. During this time the instructor/TAs will download and run your program and it will be checked for correctness using a few use-cases. If required, instructor/TA will contact you for clarifications.

Note:

If your code does not compile, it will not be graded.

Late submissions will not be accepted under any circumstances.

To be safe, always, ALWAYS, prepare to submit ahead of time, not exactly AT last moment!

Submission deadline: Friday 3 March, 11:59 PM