Climate Change Analysis

Photo Credit: kwest/Shutterstock.com

Objectives:

1. Read a data file using file I/O operations
2. Store the data in Java collection data structure(s)
3. Perform various searches on the data based on specific search criteria
4. Write the result data to another data file using file I/O operations
5. Write a short summary on your findings in climate change across the globe.

Description:

You are a weather scientist working for NOAA (National Oceanic and Atmospheric Administration; https://www.noaa.gov). You are given a data file, "world_temp_2000-2016.csv", that contains data from the Climate Change Knowledge Portal (https://climateknowledgeportal.worldbank.org) with temperature readings of various months and years for all countries. With this data, you will:

1. Write a program that asks a user to enter any required input information.
2. The program will run the climate analyzer tasks and store the results in a data file.
3. Write a short conclusion on which country has been experiencing the most drastic climate change. Support your argument with your data analysis results.

Copyright © 2020 Wendy Lee

The data that will be collected by your program will be based on the following activities:

For a specific country search (Task A) :

1) Get data which has the lowest and highest temperature reading for a given month.
2) Get data for the month which has the lowest and highest temperature in a given year.
3) Get all data that falls within a specific temperature range
4) Get data for the year with the lowest temperature and for the year with the highest temperature reading

For an inclusive all countries search (all data) (Task B):

1) The top 10 countries with the lowest temperature reading and the top 10 countries with the highest temperature reading for a given month between 2000 and 2016.
2) The top 10 countries with the lowest temperature reading and the top 10 countries with the highest temperature readings between 2000 and 2016
3) List all of the countries that fall within a specific temperature range

Climate change detection (Task C):

1) The top 10 countries with the largest change in temperature in the same month between two different years (given), for example: March 2000 and March 2016. Consider absolute values such as -14C to 28C = 42C)

Given:

1. Your program will read the file given for this project, "world_temp_2000-2016.csv"
2. Manually open this file using Excel to understand how this data is stored, as you will be writing an I/O class that will read this file. Notice how the data (per row) is comma delimited.

Grading:

The following are must do actions in order to receive full credit for this assignment.

1) You are allowed to discuss ideas with your classmates, but do not share and/or copy code with each other or from the internet. If caught plagiarizing, you will automatically fail this course.
2) Implement all interfaces and use specified classes (described below). Do not change any names of the interfaces, classes or methods.
3) The content in the write data file must be all fields of Temperature object as per row (after writing to the file, verify that the data can be viewed using Excel, Numbers app, etc..)
4) All files must be part of the same package called **`ClimateChange`**.
5) Store your java source code in a jar file called cs46b_project.jar and submit it in Canvas. If your jar file does not contain the java source code, you will not get any points.
6) Also submit in Canvas **a pdf document** with your short summary of your analysis findings.
7) If your program doesn't compile, you will not get any points.
8) Put descriptive comments throughout your code.

Copyright © 2020 Wendy Lee

```
interface ITemperature {
   public String getCountry();  // get the name of the country
   public String getCountry3LetterCode();  // get the 3-letter code of the country
   public String getMonth();   // get the month
   public int getYear();   // get the year
   public double getTemperature(boolean getFahrenheit);   // get temperature; input parameter of false = return Celsius value)
}

interface IWeatherIO {
   public ArrayList<ITemperature> readDataFromFile(String fileName);
   // read all data from the weather data file

   public void writeSubjectHeaderInFile(String filename, String subject);
   // 1. write the subject header before dumping data returned from each ClimateAnalyzer method
   // 2. a subject header is to be written for each ClimateAnalyzer method call

   public void writeDataToFile(String filename, String topic, ArrayList<ITemperature> theWeatherList);
   // 1. file name should be called "taskXX_climate_info.csv" where XX will be replaced by the task id: A1, A2, etc
   // 2. use this method to store the temperature info (for each Climate Analyzer task)
   //     a) one row for each temperature data object (i.e. all fields in one row (each comma delimited)  )
   //     b) similar to the original input data file)
   // 3. temperature value should be formatted to use a maximum of 2 decimal places
   // 4. temperature field should also show the Fahrenheit value (using decimal rules above)
   //     a) the temperature field should look like i.e. 21.34(C) 70.42(F)
}

interface IClimateAnalyzer {
   public ITemperature getLowestTemp(String country, int month);
   // TASK A-1
   // for all data that matches the specified month, get the lowest temperature reading

   public ITemperature getHighestTemp(String country, int month);
   // TASK A-1
   // for all data that matches the specified month, get the highest temperature reading

   public ITemperature getLowestTemp(String country, int month, int year);
   // TASK A-2
   // for all data that matches the specified month and year, get the lowest temperature reading

   public ITemperature getHighestTemp(String country, int month, int year);
   // TASK A-2
   // for all data that matches the specified month and year, get the highest temperature reading

   public TreeSet<ITemperature> getTempWithinRange(String country,  double rangeLowTemp, double rangeHighTemp);
   // TASK A-3
   // get all temperature data that fall within the given temperature range
   // the set is sorted from lowest to highest temperature
   // input parameter values are in Celsius

   public ITemperature  getLowestTempYear(String country);
   // TASK A-4
   // 1. get the lowest temperature reading amongst all data for that country
   public ITemperature  getHighestTempYear(String country);
   // TASK A-4
   // 1. get the highest temperature reading amongst all data for that country
```

Copyright © 2020 Wendy Lee

```
    public ArrayList<ITemperature> allCountriesGetTop10LowestTemp(int month);
    // TASK B-1
    // 1. the return list is sorted from lowest to highest temperature

    public ArrayList<ITemperature> allCountriesGetTop10HighestTemp(int month);
    // TASK B-1
    // 1. the return list is sorted from lowest to highest temperature

    public ArrayList<ITemperature> allCountriesGetTop10LowestTemp();
    // TASK B-2
    // 1. the return list is sorted from lowest to highest temperature

    public ArrayList<ITemperature> allCountriesGetTop10HighestTemp();
    // TASK B-2
    // 1. the return list is sorted from lowest to highest temperature

    public ArrayList<ITemperature> allCountriesGetAllDataWithinTempRange(
            double lowRangeTemp,
            double highRangeTemp
    );
    // TASK B-3
    // 1. the return list is sorted from lowest to highest temperature

    public ArrayList<ITemperature> allCountriesTop10TempDelta(int month, int year);
    // TASK C-1
    // 1. the countries with the largest temperature differences (absolute value)
    // 2. the return list is sorted from lowest to highest temperature delta

    public void runClimateAnalyzer();
    // 1. This method starts the climate-change task activities
    // 2. The ClimateChange methods must be called in the order as listed in the [description section], (first with the Task A
    //     methods, second are the Task B methods, and third are the Task C methods)
    // 3. For each of the ClimateChange methods that require input parameters, this method must ask the user to
    //    enter the required information for each of the tasks.
    // 4. Each ClimateAnalyzer method returns data, so the data results must be written to data files

}
```

Suggestions:

1. Start off small by coding in parts. For example, get the file I/O working first and ensure that it is working well before implementing the other classes and methods.
2. To work with implementing an interface without implementing all of the methods entirely, just so that your classes can compile, first create very minimal logic for each implementation of the interface methods. Then come back to those methods to fully implement them once you are ready to tackle that method with actual code that does something.
3. When comparing two Temperature objects, if there is a tie with the temperature reading value, then first go by the country name (i.e. 'a' comes before 'q' so 'a' wins the tie), if both objects have the same country name then go by year (earliest year wins the tie), and if year is the same for both, then go by month (the earliest month wins).

Copyright © 2020 Wendy Lee