Assignment 2

Tools & Techniques for Large-Scale Data Analytics (CT5105)

NUI Galway, Academic year 2016/2017, Semester 1

- ***Submission deadline (strict): Thursday, 13th October, 23:59***
- *Please put your code into a single .zip archive with name "YourName_Assignment2.zip", submit via Blackboard*
- *Include all source code files (that is, files with name ending .java) required to compile and run your code, including all classes, interfaces, etc.*
- *Unless specified otherwise, use only plain Java 8 for this assignment (no external libraries or frameworks)*
- *Please note that all submissions will be checked for plagiarism*
- ***Use comments to explain your source code. Missing or insufficient comments can lead to mark deductions***

Question 1 [max. marks: 40]

Suppose you want to analyze the data produced by a couple of weather stations (meteorological stations). Create a class `WeatherStation` with three attributes (fields): the *`city`* where the station is located, the station's *`measurements`* (a list of objects of class `Measurement`), and a *static* field *`stations`* (a list of all existing weather stations – you need this list only for the next question). Also create a class `Measurement`. Objects of class `Measurement` should have attributes *`time`* (an integer, representing the time of the measurement) and *`temperature`* (a double number).

Add a method `averageTemperature(startTime, endTime)` to this class which returns the average temperature measured by the weather station between `startTime` and `endTime`.

Important: implement this method using Java 8 stream operations, as far as possible (but no need for parallelism).
Also, add a `main`-method where you call your method using some test data and print the result.

Question 2 [max. marks: 60. Maximum Q2 marks if Q2 is approached without Java 8 streams: 40]

Add a method `countTemperature(t)` to your class `WeatherStation` from the previous question. It should return the number of times the temperature `t` has been approximately measured so far by any of the weather stations in *`stations`* ("approximately" means `t` +/- 1). For example, if there are two weather station objects in list *`stations`* and the first station has measured 20.0, 11.7, -5.4, 18.7 and the second station's measurements are 8.4, 19.2, 7.2, then `countTemperature(19.0)` should return 3.

For this question, you need to use an "emulated" MapReduce approach to compute the result: your program doesn't have to perform real MapReduce using a MapReduce framework (don't use, e.g., Hadoop or multiple PCs), just create code which resembles the MapReduce approach described in the lecture as closely as possible using mere Java 8 (including the use of (key, value)-pairs and parallelism!) – even where it seems "overkill" for such a simple task.

Hints: studying the "word count" example from the lecture might be helpful. Lines of text might correspond to weather stations (or cities), words might correspond to temperatures… But Q2 is simpler, since you need to count only the number of occurrences of a single temperature (that is, t).
There are several ways to approach Q2 technically. If you like, you can use only ordinary lists, arrays or array lists for this question, but you can also work with Java 8 stream / parallel stream-operations where possible. Using Java 8 (parallel) streams (where appropriate) is more elegant, but perhaps more challenging, and a good exercise for future assignments, so the maximum reachable mark on Q2 is somewhat lower if they are not used - in a sensible way - to solve this question.

Also add a `main`-method again where you call your `countTemperature` method using a few test stations and data, and print the result.