Assignment 3 (2nd half of semester)

Object-Oriented Programming III (CT3535), Year 2017/2018, Semester 1

- ***Submission deadline (strict): Saturday, 25th November, 23:59.*** *No late submission without medical certificate!*
- *Submission via Blackboard / Turnitin. Please note that all submissions will be checked for plagiarism.*
- *Put your Java source code for all questions in this assignment into a single PDF document which you then upload.*
- *Also insert into this document screenshots of any output your program(s) generate.*
- *Use Java comments to explain your code. Missing or insufficient comments may lead to mark deductions.*
- *You need **Java ≥ 8** for this assignment. Java 8 API documentation: https://docs.oracle.com/javase/8/docs/api/*

Question

Suppose you want to analyze the temperature data measured by a number of weather stations (meteorological stations).

a) Create a class `WeatherStation` with the following attributes: a non-static field *`city`* (the location of the station), a non-static field *`measurements`* (the station's measurements, as an `ArrayList` of objects of class `Measurement` - see below), and a *static* field *`stations`* (a list of all existing weather stations).

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)[1].

Add a non-static method `avgTemperature(startTime, endTime)` to `WeatherStation` which returns the average temperature measured between `startTime` and `endTime` by the weather station on which the method is called.

Implement method `avgTemperature` using Java 8 Stream operations, as far as possible. Use lambda expressions to parameterize Java 8 Stream operations, where possible.

Also, add a `main`-method to class `WeatherStation` which creates three weather station objects (each with a few fictitious measurements), adds them to *`stations`*, calls your averaging method on each of them (for some arbitrary temperature ranges), and prints the results.

Hint: There is only a small amount of code required for this question.

[max. 60 marks]

b) Add a static method `avgTemperatureAcrossAllStations(startTime, endTime)` to class `WeatherStation` which returns the average of all temperatures measured between `startTime` and `endTime`, taking into account all measurements of all stations in field `stations`.

Implement method `avgTemperatureAcrossAllStations` using Java 8 Stream operations, as far as possible. Use lambda expressions to parameterize Java 8 Stream operations, where possible.

Extend the `main`-method of class `WeatherStation` so that it calls your new method (for some arbitrary temperature range) and prints the results.

[max. 40 marks]

[1] The units of temperature and time don't matter for this assignment.