Delta College

CST 183
Programming Assignment 6

Fall 2019
Instructor: T. Klingler

Objective

To build a complete working Java program that offer practice with basic Java graphical user interface components and the interaction of a GUI with an object of a class.

Overview & Instruction

Write a Java program that will determine the risk of severe weather at a given weather station by utilizing measurements taken from a weather balloon. Weather balloons are used to observe *upper air* measurements.

There are several severe weather indexes used by meteorologists. Two are included in this assignment. Each include a simple arithmetic formula and are defined to calculate values based on patterns and conditions likely to produce severe weather. They offer a forecaster a quick number that can be referenced to assist in judging weather risks on a given day.

These measurements taken via weather balloons are not taken at standard heights, but instead at standard *pressure levels* (in the unit of *millibars*). The severe weather indexes your program will calculate require the following values:°C

T_{850}	temperature at 850 mb	T_{700}	temperature at 700 mb	T_{500}	temperature at 500 mb
Td_{850}	dew point at 850 mb	Td_{700}	dew point at 700 mb		

For validation, assume that 850 mb values must be between -40 °C and 40 °C, 700 mb values must be between –60 °C and 10 °C, and 500 mb values must be between –50 °C and 0 °C. Note also that the dew point values can never exceed the temperature values.

Below are two indexes your program needs calculate:

Total Totals Index

$$TT = T_{850} + Td_{850} - 2\,(T_{500}) \quad \text{in degrees Celsius}$$

The value produced then can be interpreted to produce the following forecasts:

Total Totals Index	Severe Weather Risk
Less Than 43	Thunderstorms Unlikely
44 to 45	Isolated Moderate Thunderstorms
46 to 47	Scattered Moderate, Few Heavy Thunderstorms
48 to 49	Scattered Moderate, Few Heavy, Isolated Severe Thunderstorms
50 to 51	Scattered Heavy, Few Severe Thunderstorms, Isolated Tornadoes
52 to 55	Scattered to Numerous Heavy, Few to Scattered Severe Thunderstorms, Isolated Tornadoes
Greater Than 55	Numerous Heavy, Scattered Severe Thunderstorms, Few to Scattered Tornadoes

K-Index

$$K = T_{850} + Td_{850} - T_{500} - (T_{700} - Td_{700})$$ in degrees Celsius

Total Totals Index	Severe Weather Risk
Less Than 20	Thunderstorms Unlikely
20 to 25	Isolated Thunderstorms
26 to 30	40% - 60% chance of thunderstorms
31 to 35	60% - 80% chance of thunderstorms, some severe
36 to 40	80% - 90% chance of heavy thunderstorms, some severe
Greater than 40	Almost 100% chance for thunderstorms, some severe

Create a class titled **`UpperAirData`** or something comparable. In this class, include the aforementioned measurements necessary for the severe weather indexes. Implement the formulas for both the Totals Index and the K-Index also in this class. Include a method to validate the data verify that the entered numbers are OK. Be sure to include at least one constructor as well as set/get methods for all variables (even though you may not need to use them for this solution). Finally, include a `toString()` method in the class.

Your solution should include **two files**: one containing the **`UpperAirData`** class including the data and method definitions and a second file to contain the "driver" application. Your driver application should include a basic graphical user interface with text fields (including appropriate labels), a text area, and a button. You will need to build a "front-end" that will allow a user to enter all of the requisite upper air observations. This implies a text field and label for each parameter. You will also need buttons to "Calculate", "Clear", and "Quit".

When the user clicks the "Calculate" button ...

1. Collect all info the input text fields
2. Set the values into your one **`UpperAirData`** object.
3. Validate the data to determine if all values are in range (from specifications above).
4. If not valid, display an appropriate message in the text area. Otherwise if valid, calculate both severe weather index values. Display the derived numbers and the corresponding messages in the text area.

The "Clear" button should empty and clear all fields. The "Quit" button should terminate the application.

To utilize actual test data, visit: http://weather.uwyo.edu/upperair/sounding.html and click on **DTX** (Detroit) to access the most recent weather balloon data for our area. Use this how-to to read and interpret the page.

Otherwise, consider this case study for a severe weather event:

T_{850} = 20 °C, Td_{850} = 15 °C, T_{700} = 3 °C, Td_{700} = 1 °C, and T_{500} = −9 °C

Deliverables

<u>Deliver</u> the following to the online course management system **dropbox** as your final product:

- **<u>Upload</u>** your **source code** (.java) files

Notice

This is an individual assignment. You must complete this assignment on your own. You may not discuss your work in detail with anyone except the instructor. You may not acquire from any source (e.g., another student or an internet site), a partial or complete solution to a problem or project that has been assigned. You may not show another student your solution to an assignment. You may not have another person (current student, former student, tutor, friend, anyone) "walk you through" how to solve the assignment.