CST 283
Programming Assignment 5

Winter 2024
Instructor: T. Klingler

Objective

This program will provide an opportunity to practice using multiple Java classes and class aggregation.

Overview & Instructions

The Department of Homeland Security, the National Weather Service, and FEMA all require a program that will provide advisory information in the event of a weather or national security emergency. Your program will receive a file of information in this form:

```
<countyFIPScode> , <startDateANDtime> , <endDateANDtime> , <warningCode>
```

For each line, your program should process the coded information and provide a clear-text message explaining the warning. Two examples are provided below:

For input:

```
26111,202002121300,202002131200,WWS
11001,202007010000,202007112359,YELLOW
```

Produce the following output:

```
Winter Storm Warning for Midland County, MI
February 12, 2020 1:00pm - February 13, 2020 12:00 pm
Population Impact: 83,919

Significant risk of terrorist attacks for District of Columbia, DC
July 1, 2020 12:00 am - July 11, 2020 11:59 pm
Population Impact: 646,449
```

The government has created a dataset of *FIPS* codes. Each county in the U.S. is assigned a unique five-digit code. This code is used to identify the county when alerts are generated. It is most used for weather warnings, but also could be used for industrial or national emergencies. For example, for Saginaw County, MI, the code is 26145. (Note: the 26 is Michigan's state code and the 145 is the unique county identifier within the state.)

Features to note:

- Times are presented in a 24-hour military format. Be sure to decode to a traditional format that includes am/pm
- Add a feature that will insert commas as needed to represent the *periods* (groups of three) for larger numbers

Design your solution using an object-oriented approach. The following classes are expected or this assignment:

Class	Purpose
`County`	Data for one county: FIPS code, population, and name.
`CountyList`	List of County objects. Include ability to search by FIPS code and return county object.
`Alert`	One alert. Includes date/time, code, and all data related to one alert. Includes functionality necessary to write alert description.
`AlertList`	Basic class to store, manage and sort a list of alert objects. Recommended to own and manage an instance of the `CountyList` class for county data referencing.
`AlertProcessor`	Driver class with high-level actions utilizing one `AlertList` object.

Optionally, you could also add a class to manage the calendar date. This is not required but the necessary coded date conversions are typically seen in applications and would be a nice addition to the existing class used in the course.

Your program should be driven by a list of warning messages (found in file alerts.txt). Each line of data in this file represents a distinct warning message that is disseminated by an agency of the U.S. Government. Ultimately the output should be sorted with (1) security warnings at the top, and then (2) weather alerts to follow in the order of warnings, then watches, and finally advisories.

Files that will be necessary for this solution:

fipsCounty.txt	Data file with county names associated to county FIPS codes
popCounty.txt.zip	Data file with county populations by FIPS code
warningList.txt	Specifications file with list of warnings by code

You can download a compressed set of all files using this link.

The `warningList.txt` file is your reference for the warning messages and is provided in a less than useful format. You are free to reformat that in any way you wish, but please keep this as an external reference (i.e. avoid hardcoding the content in the program code, but instead reorganize it for easier reading via file input).

A user interface is not necessary for this program. Output can be displayed via the basic Java console. Finally, please avoid use of `ArrayList` or any other built-in Java "collections" data structure.

Deliverables

Demonstrate the development steps of your program with at least two version commits to the assignment Git repository.

Deliver the following to the eLearning system **Assignment Dropbox** as your final product:

- **Upload** your **source code** (.java) file(s); preferably as one zipped submission