CECS 277 – Lab 3 – File IO

State Stats - Create a program that displays statistical information about US states and territories and their populations. The data for the states is stored in the file 'StatePops.txt'. Assume you do not know how many states are in the file (ie. your program should work with any length file). Use a pair of parallel ArrayLists to store the states (Strings) and their populations (Integers).

Create the following methods:

1. readFile – pass in both ArrayLists, read the values from the file and add the states to one list and the populations to the other (be sure to keep them in the same order).
2. sortStates – pass in both ArrayLists, sort the parallel lists alphabetically by state name (note – you can't compare strings using relational operators, only string functions).
3. sortPopulation – pass in both ArrayLists, sort the lists by population (from low-high).
4. displayStates – pass in both ArrayLists, display each state and its population.
5. totalSum – pass in the populations ArrayList, calculate and return the sum.
6. populationGreater – pass in both ArrayLists, prompt the user to enter a population, then display all states and their populations that have a larger population than what the user entered.
7. menu – display the menu options below.

Menu – Repeat the menu until the user quits.

```
State Stats
1. Display Sorted States
2. Display Sorted Populations
3. Display Total US Population
4. Display States with Population Greater Than
5. Quit
```

Notes:

1. Do not throw exceptions from main. Handle the exception in your readFile method.
2. Use a pair of ArrayLists, not arrays.
3. Read in the file when the program starts (readFile should only be called once).
4. Do not use global variables.
5. Display all populations formatted with the thousand's separator (ex. State 10,987,654) (See Lecture 1B).
6. Check all user input for validity. Use the CheckInput class given on Beachboard.
7. Use Javadocs to document all functions with a description, parameters, and return values. Add brief comments to describe your code.
8. Place your names, the date, and a description of the program in a comment block at the top of your program.
9. You cannot use relational operators to compare strings. Instead, use the string function compareTo or compareToIgnoreCase when sorting the state names. The compareTo function returns a 0 if the two strings are the same, a negative value if they are in order, and a positive value if they are out of order.

   Ex. `if( list.get(i).compareTo( list.get(i + 1) ) > 0 ) {`
10. Use the following sorting algorithm for your sorts. You will need to modify it to work for your program (ie. change it to strings for sorting the state names, and/or add in the

extra swaps for the parallel array). You won't be able to use Collections.sort(), it will not work for the parallel arrays.

```
public static void sort( ArrayList<Integer> list ) {
      boolean swapped = false;
      do {
            swapped = false;
            for( int i = 0; i < list.size() - 1; i++ ) {
                  if( list.get( i ) > list.get( i + 1 ) ) {
                        int swap = list.get( i );
                        list.set( i, list.get( i + 1 ) );
                        list.set( i + 1, swap );
                        swapped = true;
                  }
            }
      } while( swapped );
}
```

Example Output:

```
State Stats
1. Display Sorted States
2. Display Sorted Populations
3. Display Total US Population
4. Display States with Population Greater Than
5. Quit
1
Alabama 4,903,185
Alaska 731,545
American Samoa 55,641
Arizona 7,278,717
Arkansas 3,017,825
California 39,512,223
Colorado 5,758,736
...
State Stats
1. Display Sorted States
2. Display Sorted Populations
3. Display Total US Population
4. Display States with Population Greater Than
5. Quit
3
US Population = 331,875,705
State Stats
1. Display Sorted States
2. Display Sorted Populations
3. Display Total US Population
4. Display States with Population Greater Than
5. Quit
4
Enter Population: 25000000
Texas 28,995,881
California 39,512,223
```