CSCD 211
Lab 1

Provided is CSCD211Lab1.java that is an unchangeable Java file. This file resides in the package cscd211Lab1. Your task is to complete the program based on the specifications below.

- You will prompt the user for the name of the input file. The file specifications are below.
- You will open that file and count the number of records in the file
- You will create and fill an array of type person for each record that is in the file.
- The array is then sorted by calling the compareTo of Person
- A menu is displayed – specifications are below
    - Specific methods based on the menu choice

File Specifications

Information on each individual will be in the following format (all lines in the file will end with a carriage return):

First name
Last name
Favorite color

Example
bubba
von bigbelly
green
sally sue
sullivan
purple
faith
hillstrom
purple

The information from the file will be stored in an array of type Person. The information about a Person is specified below

Class Person

Private Data Members

- String for first name
- String for the last name
- Enumerated Type Colors for the color

Public Methods

- compareTo that sorts by last name if the same then first name if the same then color– must ensure a Person is passed in at Compile Time
- toString that returns first name last name – color NOTE: information will be capitalized. (Example: Bubba Von Bigbelly – Purple)
- Explicit Value Constructor – takes 2 Strings and an enumerated type
- Get and Set methods as you deem necessary
- Other methods you deem necessary

Enumerated Type Color

Constant values for

- red,green,yellow,purple,blue,pink - the lowest value is red, then green, then yellow, purple, blue and pink, is the highest value
- You must have a private explicit value constructor that is passed a value for each color.
- A toString the returns the appropriate color all lower case
- A getValue that returns the underlying value

CSCD211Lab1Methods

- fillArray – creates and fills the array by reading from the file
- Menu that has the following choices
  1. Print the Array to the screen calling the toString
  2. Display all people that contain a certain color
  3. Sort the array by Color only using the Comparator
  4. Sort the array by the "natural order" using Comparable
  5. Quit

- displayAll – Prints all person information for a specific color to the screen in the format of first name space last name – color (Example: Bubba Von Bigbelly – Purple) Note: the capitalization
- printArray – Prints the information in the array calling the toString – one person per line.
- convertColor – receives a String and converts that string to the proper Colors type and returns it – method is private.
- readColor – prompts the user to enter a color, ensures the color is valid and returns the appropriate enumerated data type for that color.

NOTES

- You must range check all values
- You may assume all letters will be lower case for any input obtained by your program
- You must ensure the parameterized type of Person is passed to compare/compareTo
- You need to capitalize the name when you print out the name
- All parameters will be final and all preconditions will be checked
- A jar file containing the FileUtil and SortUtil methods is provided and must be used.
- You must create an input file that allows you to sufficiently test the specifications of this assignment.
- I have provided the Javadocs from the solution. Please see index-all.html inside the docs folder.

TO TURN IN

There are multiple turn ins for this lab.

1. Your stubbed out methods ensuring all code compiles will be submitted via git commit to your repository on GitHub. This is required within by 11 PM on January 6, 2016. You need to become very familiar with incremental builds and version control. GitHub allows you to maintain working code in a simple version control system.

2. A zip file, in Canvas, by the required due date containing:
   - all java files
   - the input file(s) used to test your program program
   - a test run named cscd211lab1out.txt – testing all aspects
   - We should be able to download the zip and compile your code, and then run your code.
   - Name your zip, your last name first letter of your first name lab1.zip (Example: steinerslab1.zip)