CIT 591 – Homework 4

Due – Nov 7, 2017 at 12.00pm

Part 1 – Theory (20 points)

Please do the following REVIEW EXERCISES from the class textbook:

1. R7.9, R7.19, R7.32, R11.1 (5 points each)

Part 2 – Philadelphia Bike Share Data and Jeopardy (80 points)

Analyzing Data from the Internet: Real-world data sets are becoming increasingly available for a large variety of domains. E.g., www.data.gov has datasets ranging from education and agriculture to manufacturing and energy. https://www.opendataphilly.org/ provides a lot of datasets for the Philadelphia region.

For this assignment, we will use the Indego Bike Share Data on bike rentals and trips in Philadelphia. You can download the datasets and read more about what's available here https://www.rideindego.com/about/data/. In particular, you'll need to download and use the following two files:

1. Trip Data for the third quarter of 2017 - https://u626n26h74f16ig1p3pt0f2g-wpengine.netdna-ssl.com/wp-content/uploads/2015/12/indego-trips-2017-q3.csv.zip
2. Station Table - https://www.rideindego.com/wp-content/uploads/2015/12/indego-stations-2017-10-20.csv

You'll write a program in Java to read, analyze, and summarize this information Once you have a basic program that can read in the data files, use it to answer the following questions. Describe in detail the algorithm you used for each question and the answers in your readme.txt file.

Analysis of Data

1. How many *One Way* trips were there in the third quarter of *2017*?
2. How many stations that had a Go-Live Date in *2016* are still *Active?*
3. What percentage of trips ended at the *Philadelphia Zoo*?
4. In which month were the most *Indego30* trips taken?
5. What is the ID of the bike that has traveled the most in terms of duration?
6. What percentage of trips happened between *12.00am* (midnight) and *5am*?
7. On *9/15/17* at *7:00am*, how many bikes were being used?
8. Print (to the console) all the trip information for the longest trip by distance. To make things easier (and not worry about spherical geometry), we'll assume that the latitude and longitude are points in 2-d space and use Euclidean distance.

9. Print (to the console) the total count of the number of trips that involved a station which was the only station to go live on its respective go-live date.
10. Wild card – come up with an interesting question. List the question and find the answer to it.

Summary of Data by Station

Finally, you need to provide a summary of data grouped by station. Create a file in Java and write out to it in the following format:

- There will be one line for each station in the file
- Each line will have the following format: <station id>, <station name>, <total number of trips>, <average trip duration (in terms of time) from this station>, <average trip distance (in terms of Euclidean distance) from this station>, <max trip duration (in terms of time) from this station>, <max trip distance (in terms of Euclidean distance) from this station>, <percentage of one way trips>, <difference between the total number of trips that start at this station and that end at this station>

Software Design: An important part of this homework is designing your classes and methods. Using the “Nouns and Verbs” approach and keeping good design principles in mind, create CRCs. This should be done *before* you start implementing your code in Java and you need to submit the original CRCs as part of the submission.

When you start implementing your code, it’s perfectly fine to go back and change the design, if needed. You should explain what you changed and why in the `readme.txt` file.

Note: For the italicized parts in the above, your code should be able to deal with any similar input (e.g., from a user). This should not be hard coded.

Part 2 – Extra Credit (20 points)

In addition to the questions above, answer the following questions:

1. Stations can be located using the longitude and latitude coordinates provided in the dataset. We can define “closeness” as follows: Two stations are considered close to each other if the average difference between their longitudes and latitudes, i.e., (difference longitude + difference latitude) / 2, is less than *0.02 points*. Find all pairs of stations that are considered close to each other.
2. What is the *least* popular *end station*?
3. Wild card – come up with an interesting question. List the question and find the answer to it.

As before, for the EC part, you cannot have any help from the TAs/instructor.

Grading Criteria (for the Programming part)

5% for compilation – If your code compiles, you get full credit. If not, you get a 0.
65% for functionality – Does the code work as required? Does it crash while running? Are there bugs? …
15% for style – Do you have good variable names? Is your code well commented? …
15% for design – Is your code well designed? Do you follow good software design principles?

Programming – General Comments

Here are some guidelines with respect to programming style.

Please use Javadoc-style comments.

For things like naming conventions, please see Appendix I (Page A-79) of the Horstmann book. You can also install the Checkstyle plugin (http://eclipse-cs.sourceforge.net/) in Eclipse, which will automatically warn you about style violations.

Submission Instructions

We recommend submitting the theory part electronically also. However, you can turn in a physical copy at the start of class, if you prefer. Please **do not** print out the Java source.

In addition to the theory writeup, you should also submit a text file titled readme.txt. That is, write in plain English and instructions for using your software. You should also include explanations/rationale for why you chose to design your code the way you did and whether you deviated from your original design and why. The readme.txt file is also an opportunity for you to get partial credit when certain requirements of the assignment are not met. Think of the readme as a combination of instructions for the user and a chance for you to get partial credit.

Please create a folder called YOUR_PENNKEY. Places all your files inside this – theory writeup, the Java files, the CRCs, the readme.txt file, the ec.txt file. Zip up this folder. It will thus be called YOUR_PENNKEY.zip. So, e.g., my homework submission would be swapneel.zip. Please submit this zip file via canvas.