CS 150: Project 2
Fall 2015

Version: Tuesday 27th October, 2015–22:04

Due: November 14, 2015

Project Description

The goal of this program is to help identify and develop effective advertising outlets for the Easton Farmers market, and for the vendors who are part of the market. The key to doing this is to be able to efficiently and effectively provide information about people, and their individual interests.

Program Behavior

This program will be an extension of your program from Project 1. Your program will read in a text file of city names (*cities.txt*) where each line is the name of a city. Every customer will be randomly assigned a city as their residence. Each advertising outlet obtains its revenue from one or more cities and one or more types of vendors. The. program will be used to help determine where advertising can be used most effectively.

Commands

The input to your program (from the console) will consist of a series of commands to update/modify/query the data in your program. The commands are:

- **init**: will read the file *cities.txt* and store the information appropriately
- **run** $< n >$): will run your simulation n times and add the data from your simulation to the appropriate containers
- **clear**: will clear all the information from the containers
- **list:** followed by lines of text. The input terminates with an empty line. Each line has the format of:
  - city: <name of city> or
  - item: <name of item>

  The program will print the names of the individuals who live in one of the specified cities and who have bought **ALL** of the specified items at the farmers market.

- **findgoods**: followed by lines of text. The input terminates with an empty line. Each line has the name of a city.

  This command will print the list of goods in order of most purchased to least purchased by the people from the cities.

- **findcities**: followed by lines of text. The input terminates with an empty line. Each line contains the name of one good. The command will print the cities sorted by the number of people from those cities who purchased that combination of goods.

- **add:** followed by one line of text. The line has the format of:

  - city: <name of city> or
  - item: <name of item>

  This adds the city to the container of cities that people come from. This affects subsequent generation of data.

- **exit**:

Constraints

Your program should use (in an appropriate manner) all the data structures that we have covered so far. They are:

- ArrayList
- List - you can use a List, or a Stack or a Queue
- Hash Map/Set
- Tree Map/Set

Restrictions

This project is to be done individually. You may discuss concepts and ideas with other students but you can only discuss programming issues with the instructor.

Report

Some questions that you might want to answer about your program:

1. correctness of your program - what functionality does your program implement correctly (or partially)?

2. performance of your program - how does the performance vary for each command as the size of the database grows? The goal of the program is to have the best performance (complexity) for answering the queries/commands. You should discuss the tradeoffs/design choices you made and how they affect the performance of the queries/commands.

Grading

Your project will be graded on the following criteria (assuming the program compiles and runs):

1. correctness of the program
2. documentation (methods and classes)
3. unit testing
4. object oriented design
5. quality of the simulation and analysis

Submission

Submission is in 3 parts:

1. Part I - 2pts (due Saturday Oct 31): the ”story” of your program
2. Part II - 2 pts (due Saturday Nov 7): class diagrams for your program
3. Part III - 96 pts (due Saturday, Nov 14): the rest including a draft report

Your submission for Part III will be composed of the following:

1. source files (*.java) that are commented and have javadoc directives
2. test files, one test file per class
3. a README.txt file that contains instructions on how to run your program
4. a draft of the project report (see project report guidelines `www.cs.lafayette.edu/~liew/courses/cs150/writeup-guidelines.pdf`) - 50% of the report grade is assigned to this. The remaining 50% will be given to a final report. The final report will be due several days after the draft is corrected and returned.

Additional Notes

Your program does not have to run the actual simulations from project 1. You will only need to generate customers with the appropriate needs with the addition of the city of residence. Note that all the probability distributions (mean, std deviation) that were used for project 1 remain unchanged. In particular, this means that each run should generate a different number of customers based on the interval between their arrivals.