CS 150: Project 2
Spring 2015

Version: Thursday 2nd April, 2015–13:48

Due: 11:55pm April 12, 2015

Project Description

For this project, your task is to design, implement and evaluate an electronic cookbook. The cookbook will be used to organize recipes and prepare meals. The supported operations for the cookbook are:

1. add a recipe
2. delete a recipe
3. find a recipe
4. plan a meal

Program Behavior

Your program will read in a text file of recipes, create the appropriate storage and indices to hold the information and then repeatedly query the user for commands.

Inputs

The inputs to the program are composed of:

- a text file (*recipes.txt*) that contains the recipes of interest. Each recipe is separated from the next by a blank line. A recipe is described by a series of lines, each of which consists of a keyword value pair. The value may consist of multiple words separated by a space. The keywords and legal values are:
  - cuisine: Italian,Chinese,Greek,Turkish,Indian,Pakistan,French,Korean
  - name: <made up>
  - prepTime: <integer>
  - cookTime: <integer>

  - type: salad,entree,appetizer

    Each type of dish will have the following information:

    * entree
      - main: beef,chicken,lamb,goat,salmon,tofu,tempeh,shrimp,scallops
      - addons: broccoli,cauliflower,spinach,mushrooms,beans,potatoes,beets,peas,rice
      - sides: eggs,cheese,milk,peas,beans
    * salad
      - main: kale,spinach,lettuce,chard,mesclun,arugula
      - addons: radishes,scallions,eggs,cheese
      - sides: cucumbers,carrots,mushrooms,potatoes
    * appetizer
      - main: beef,chicken,tofu,shrimp,scallops
      - addons: peas,rice,cheese,eggs
      - sides: spinach,mushrooms,cheese,beans

- commands entered through the console/terminal. Prompts are recommended. The valid commands are:

  - add - add a recipe. Prompt the user (one line for each value) for the required information.
  - delete - prompt the user for a name, display the recipe if found and prompt for confirmation. If the recipe is not found, display a one line error message.
  - find - prompt the user for the appropriate information, display the recipe(s) if found. If no recipes are found, display a one line error message. The information that is to be supplied is:
    * totalTime: meal preparation cannot overlap but cooking the meal can overlap.
    * requiredItems: any of the ingredients listed above (in the description) and includes the following additional words - seafood,meat,dairy,vegan,vegetarian,shellfish
    * excludedItems: any of the ingredients listed above (in the description) and includes the following additional words - seafood,meat,dairy,vegan,vegetarian,shellfish
    * cuisine: any of the cuisines listed above (in the description) and includes the following additional words - Asian,MiddleEastern,SouthAsian
    * definitions:
      - vegetarian - no meat, no seafood
      - vegan - vegetarian, no dairy, no eggs
      - Asian - Chinese or Korean
      - MiddleEastern - Greek or Turkish
      - SouthAsian - Indian or Pakistan

  - plan - prompt the user for the appropriate information, generate an appropriate meal plan that consists of 1 entree, 1 salad and 2 appetizers. Display a one line error message if the plan cannot be generated. Display the plan that comes closest (and less) to the time given. If there is more than one plan with the same time, display all the matching plans.
  - quit - write the recipes to a new file (*newRecipes.txt*) and exit. The new file should be in the same format as the input file.

Outputs

The outputs of the program will consist of the following in response to each command

- add - one line confirmation of addition of recipe or a one line error message if the name already exists
- delete - display the recipe and confirmation of deletion or a one line error message if the recipe cannot be found
- find - display the recipe or a one line error message if the recipe cannot be found
- plan - display the components of the meal, with a blank line separating the components.

Constraints

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

- ArrayList
- List - this includes Stack and Queue
- Hash Map/Set
- Tree Map/Set

Analysis

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

1. correctness of your program - what functionality does your program implement correctly? How well does your program do at satisfying the time limits for a meal plan?
2. performance of your program - how does the performance vary for each command as the size of the database grows?

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.