Michael Wells

Module 11 Collections Design

08/11/19

**Collections**

The first part of my program prompts the user to enter the location of the contacts txt file. I designed my program not to crash or require the file to exist so that if the file doesn’t exist you could add contacts then write to this file location. Either way I create a TreeMap object called contacts that has a string for its Key, which we will use the last name for so that it sorts by the last name, and a contact object for the value of the map. I created a contact class, its info is shown below using UML design tool.

If the contacts txt file exists, I used a dataInput stream to read in line by line the file, then parse all required strings for the contact class, construct the contact object, then put this object in my contacts tree using the getContactLastName() method to use the last name as the key for the value. Once we have loaded the contacts if they exist. The user has the option to add a contact, delete a contact, show all contacts, and exit the program.

Adding a contact prompts the user to enter all the values needed to construct the contact class using the addContact() method. Puts the contact into contacts tree, then write the file to the file location using writeUTF for each parameter of contact, for each contact in the contacts tree.

When the user wants to remove a contact, they are asked to enter the last name of the contact to remove. This is of course because we used the last name as the key to the tree so I simply take this string and use the remove() method.

To show all contacts, I iterate through the contacts tree and use the getSummary() method from the contact class to output all the information of the contact.

****