CSC240 – SPRING 2013
ASSIGNMENT #3

Generics

This assignment involves modifying the generic class Set<T> from our example programs that relate to generics (GenericsOne, GenericsTwo, GenericsThree).

Your program will create a Set that contains FrozenFood and CannedFood objects. Thus, the program will have to declare a set of Element references: Set<Element>. You may use the FrozenFood and CannedFood classes from assignment #2 modified slightly to take into account the modifications we have made to the abstract class Element. The new Element class used in the generics programs cited above **does not declare an equals method** and it contains a toString method. In addition, remember to modify the equals methods in the FrozenFood and CannedFood classes so that they have an Object parameter and not an Element parameter.

Here again are the fields for the FrozenFood class:

Field	Type	Description
name	String	Name of the frozen food product (e.g., "VEGGIE BURGER")
manufacturer	String	Name of the company that makes this frozen food product ("FRESH GARDEN")
nutrients	String	A list of the top nutrients that this frozen food provides (e.g., "PROTEIN, SODIUM; VITAMIN A")

Here again are the fields for the CannedFood class:

Field	Type	Description
name	String	Name of the CannedFood (e.g., "CHILI BEANS")
manufacturer	String	Name of the company that makes this frozen food product ("FANTASTIC BEANS")
ingredients	String	A list of the top ingredients in this canned food product (e.g., "KIDNEY BEANS; PINTO BEANS; GARABANZO BEANS")

Insofar as the generic class, Set, is concerned, you will add two new methods, similar to the methods you implemented for the ElementSet class for assignment #2. These are the displayAnObject and the editAnObject methods, described below.

Here is a description of the displayAnObject method:

Header:	public boolean displayAnObject(T aT)
Description:	This method will search for aT in the Set. If it fails to find aT it will return false and the method itself will do nothing else. The client will be responsible for giving the user feedback if aT is not found. If it finds aT, then it will display aT using the toString method (since only Object class methods can be applied to a T-thing in our generic class. If it finds aT the method will return true.

Here is a description of the editAnObject method:

Header:	public boolean editAnObject(T editedT)
Description:	This method will search an object that is equal to edited in the Set. The search criteria are the same as for displayAnObject. The method will be looking for an object that belongs to the same class as editedT and that is equal to editedT (based upon a call to the equals method). If a matching object is found, it is replaced by edited and the method returns true. If a matching object is not found, the Set remains unchanged and the method returns false.

the application program

The application program will present the user with the following menu which is identical to what we had in assignment #2:

Wholly Nutritious Food Data Menu
1 – Add a Frozen Food Product
2 – Add a Canned Food Product
3 – Display names of all Frozen Food Products
4 – Display names of all Canned Food Products
5 – Display data for a Frozen Food Product
6 – Display data for a Canned Food Product
7 – Edit data for a Frozen Food Product
8 – Edit data for a Canned Food Product
9 – Quit

You will develop an application that uses the modified FrozenFood and CannedFood classes and the modified Set<T> generic class. The application will present the user with the above menu over and over again, until the user indicates that he/she wants to quit. The descriptions for each of the menu choices is very similar to what we had for assignment #2, but they are presented here once again.

1 – Add a Frozen Food Product

Prompt the user for the FrozenFood data. If that FrozenFood object is already in the Set (as indicated by the value returned by the add method), give the user appropriate feedback. Otherwise, add the FrozenFood object to the Set and give the user appropriate feedback.

2 – Add a Canned Food Product

Same idea as adding a Frozen Food product, except we are adding a Canned Food object to the ElementSet. In all cases, let the user know what transpired (success or failure).

3 – Display Names of All Frozen Food Products
An important issue here is that ONLY the names of the Frozen Food products will be displayed, not all of the data for each of the Frozen Food objects in the Set. The names of each of the Frozen Food products in the Set will be displayed. However, if there are no Frozen Food objects in the Set, the user is given appropriate feedback.

4 – Display Names of all Canned Food Products
An important issue here is that ONLY the names of the Canned Food products will be displayed, not all of the data for each of the Canned Food objects in the Set. The names of each of the Canned Food products in the Set will be displayed. However, if there are no Canned Food objects in the Set, the user is given appropriate feedback.

5 – Display Data for a Frozen Food Product
Prompt the user for and get the name of the Frozen Food product. Stuff that name into a FrozenFood object and pass that FrozenFood object as the parameter in a call to the displayAnObject method. If an equivalent FrozenFood object is found, the data for that FrozenFood will be displayed and the displayAnObject method will return true. If that FrozenFood object is not found, the displayAnObject method will return false and the calling method in the application will be responsible for giving the user appropriate feedback.

6 – Display Data for a Canned Food Product
Same idea as displaying a Frozen Food product, but this time we will be displaying the data for a CannedFood object using a call to the displayAnObject method. If the search fails, the application will be responsible for giving the user appropriate feedback.

7 – Edit data for a Frozen Food Product
The user is asked to enter the name of the FrozenFood product. The user is then prompted to enter the revised data for the FrozenFood product. Essentially, you can just use the FrozenFood readIn method in order to accomplish this reading in process. You don't need to ask the user which FrozenFood data the user wants to modify. I made this decision in order to

keep things simple.   So, basically you are creating a FrozenFood product and getting all of the data for that product. If that product is in the ElementSet, it is replaced (using the editAnObject method) with the new data and the user gets appropriate feedback. If that product is not in the ElementSet, the user gets appropriate feedback.

8 – Edit data for a Canned Food Product

Same ideas as editing the data for a FrozenFood product. The user is asked to enter the name of the CannedFood product. The user is then prompted to enter the revised data for the CannedFood product. Essentially, you can just use the FrozenFood readIn method in order to accomplish this reading in process. You don't need to ask the user which CannedFood data the he/she wants to modify. I made this decision in order to keep things simple. So, basically you are creating a CannedFood product and getting all of the data for that product. If that product is in the ElementSet, it is replaced (using the editAnObject method) with the new data and the user gets appropriate feedback. If that product is not in the ElementSet, the user gets appropriate feedback.

9 – Quit

If the user chooses this menu choice, the user must confirm this choice (as in assignment #1). If the user confirms that he/she wants to quit, the program terminates. Otherwise, the user is taken back to the main menu.

grading

Your program will be graded using a grading sheet that will be posted on D2L. Your program should satisfy the items in the grading checklist as much as possible.   (Hopefully, 100%.)   In addition, your program will be graded on:

- quality of the documentation
- clarity of your code
- the use of appropriate conventions of style
- correct use of classes and objects
- timely submission of your work
- a filled in, submitted project plan form