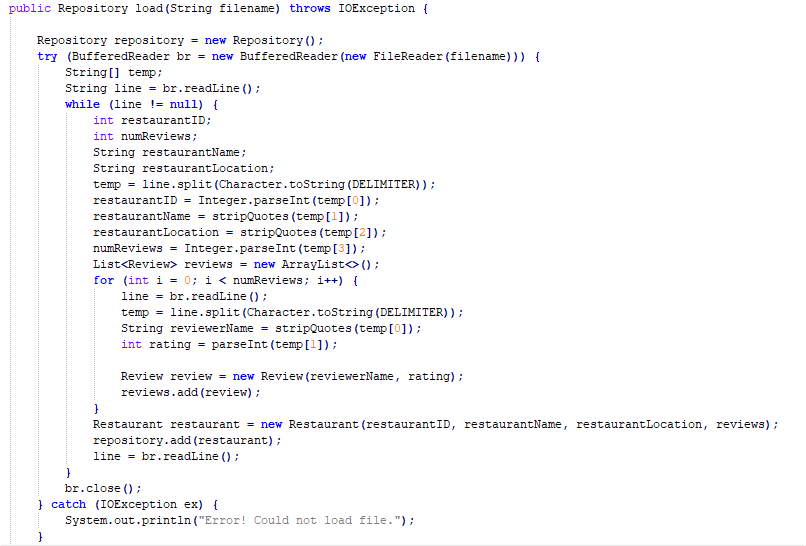
**Increment 1**

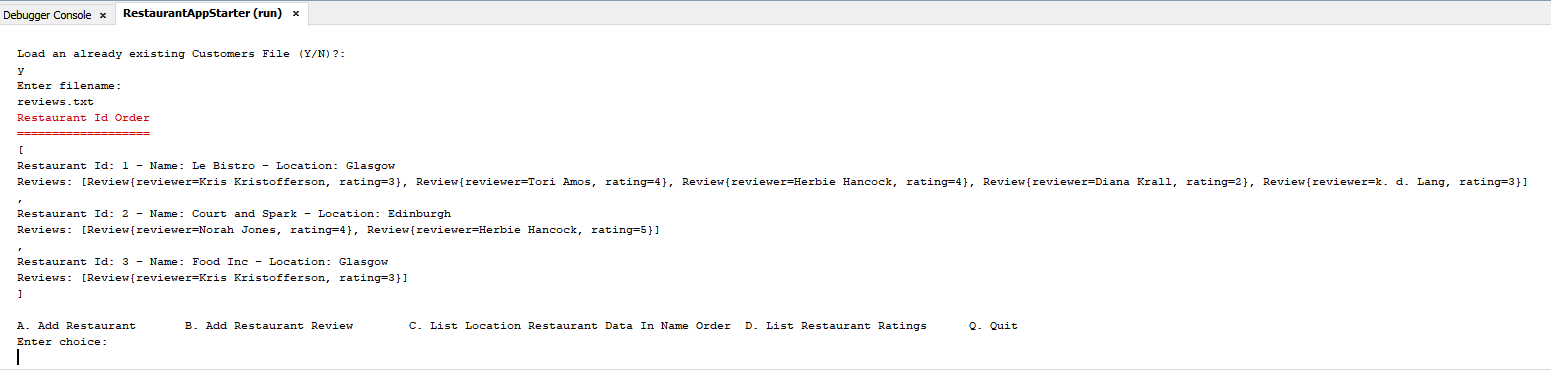
This method lists the restaurant data in ID order, smallest to biggest. The way this method works is by creating a list of restaurants called ‘repositoryItems’ and then using the ‘getItems()’ function on ‘this.repository’ and what this does is load in the text file data into a repository object. The text file name was gotten by the users input when the program was ran initially.

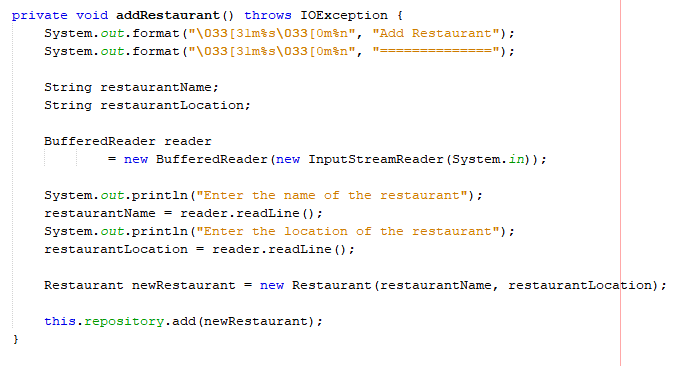


This store method gets in the filename of the file which is given by the user when starting the program. The method then loops through each line reaches the end of the file, and for each input it reads in the individual bits of data, separated by a comma, and saves each individual data piece to an according variable. To read in the reviews for each restaurant the last part of the line is the number of reviews, so a loop running that many times is created to read in the following same number of lines and each part of that line is separated by a comma, for the reviewer name and rating, and saved to the according variable name, and a review object is created using those reviews which got saved to a list of reviews.

The previous variables read in such as the restaurant ID, name, location, and now the list of reviews called ‘review’ are used to create a new Restaurant object called ‘restaurant’ and are added to a repository object and this loops until it reaches the end of the file.

If a file couldn’t be read, then an error is returned.



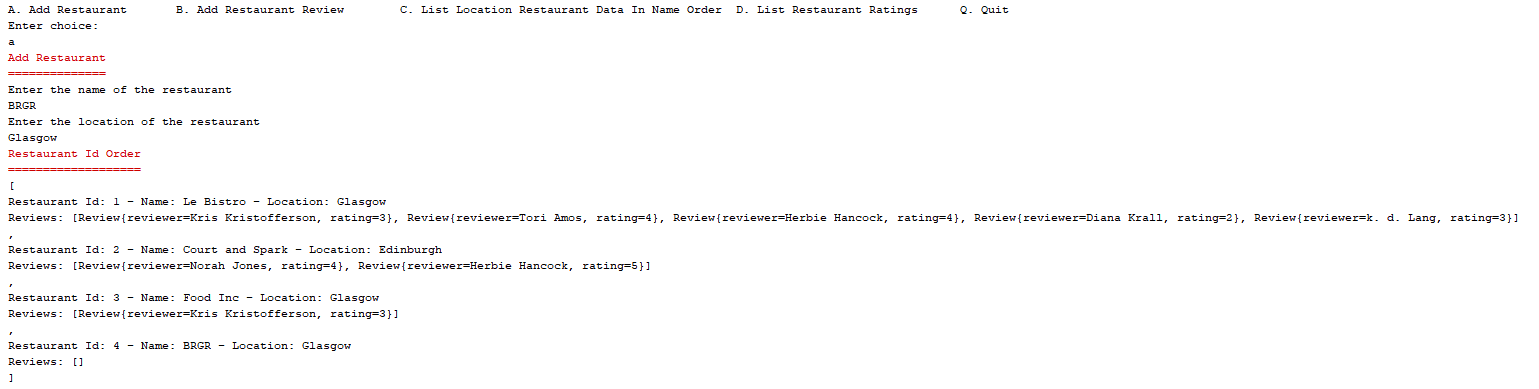
**Increment 2**

This method is adding a new restaurant to the existing repository item. How this method works is firstly a String is made to hold the restaurant name and location, named accordingly, and then a BufferedReader is created to take in the user input.

The first input taken by the user is the name of the restaurant stored in ‘restaurantName’ and then the second input is the location of the restaurant, which is stored in restaurantLocation.

Then a new restaurant object is created, named ‘newRestaurant’, and a new restaurant is created by calling the constructor Restaurant and passes in the variables restaurantName, and restaurantLocation, which are String.

This specific constructor creates a restaurant object with a new ID which is 1 more than the last allocated ID, and sets the name and location according to what the user inputted, and also creates a new empty list of reviews.



**Increment 3**

This method is to add a review to either an existing exiting restaurant.

The way this works is firstly by creating a String variable to hold the name of the reviewer called ‘reviewerName’ and two integer variables to hold the ID of the restaurant (restaurantID) and the rating (named rating).

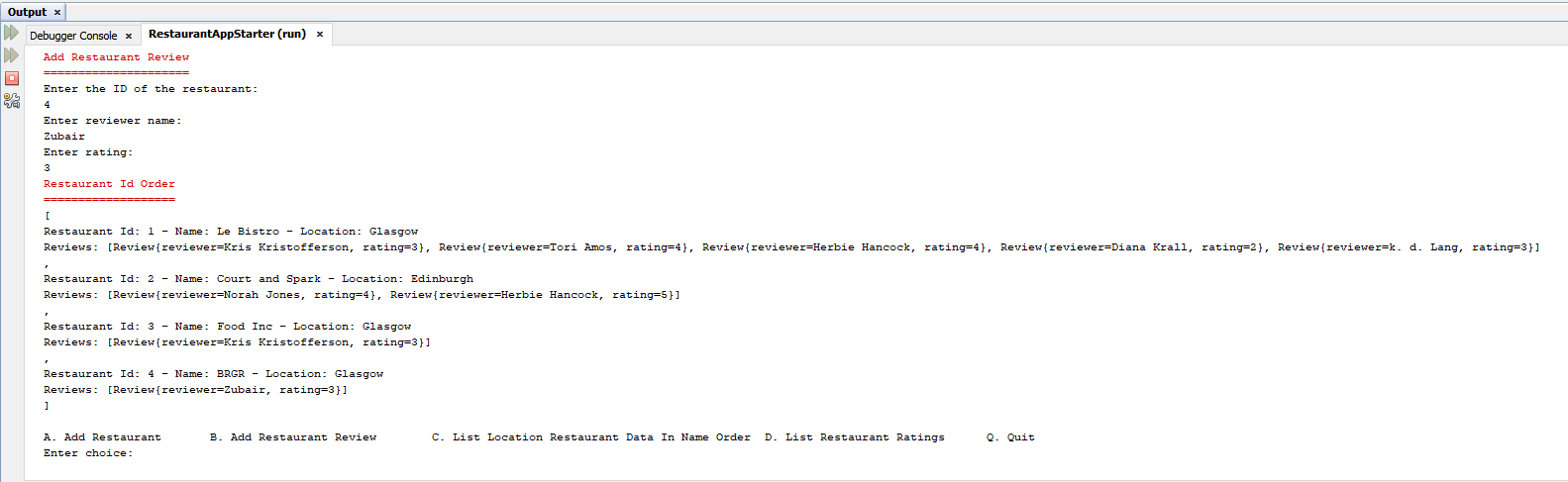
Then a BufferedReader is used to read in the users input.

The first thing read in is the ID of the restaurant and is parsed in as an integer to be stored as an int. This is done first as the user needs to know which restaurant, they’re adding the review to first before they create the actual review.

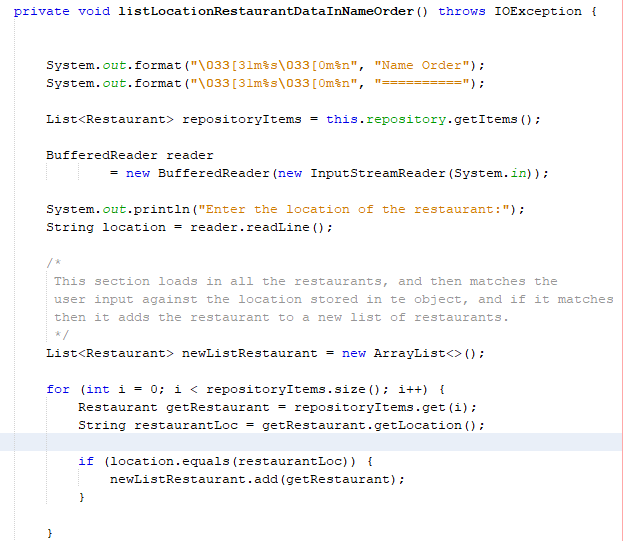
The second input is the name of the reviewer, and the third input is the rating of that review which is parsed in as an integer.

Then a new Review object is created called ‘newReview’, and it calls the Review constructor which takes in the name of the reviewer and the rating, and this creates the object which is saved to ‘newReview’.

Next the current items in the repository, which is the current restaurants that exist, is read into a new list of restaurants called repositoryItems. Then then how the review is added to the whichever according ID is by doing n-1, as the list starts from 0 for ID 1 and so on, so if the user was picking restaurant ID 4 the program would see that the 4th restaurant is 3 in the list, so 4 – 1 = 3, and then the code reads in the 3rd restaurant object in repositoryItems, and add that to a new Restaurant object called tempRestaurant, and then tempRestaurant calls its ‘addReview’ method, which passes in newReview and adds that review to that restaurant object.



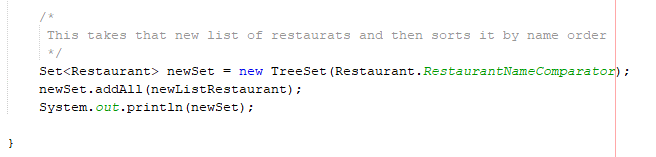
**Increment 4**

This method is to list the use the location given by the user and filter the restaurants out to list only those restaurants in that location and list those restaurants in name order.

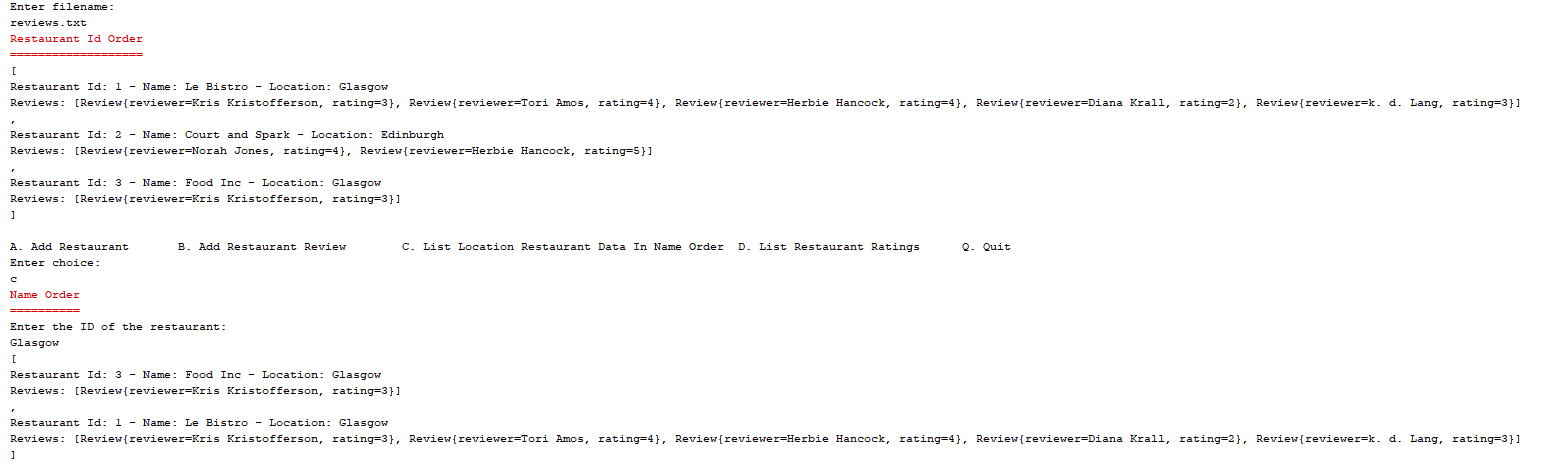
Firstly, the current items in the repository are gotten and stored into a list of restaurants called repositoryItems.

A new BufferedReader is created to take in the user input, and the input that is read in is the location of the restaurant and stores this to a String variable called location.

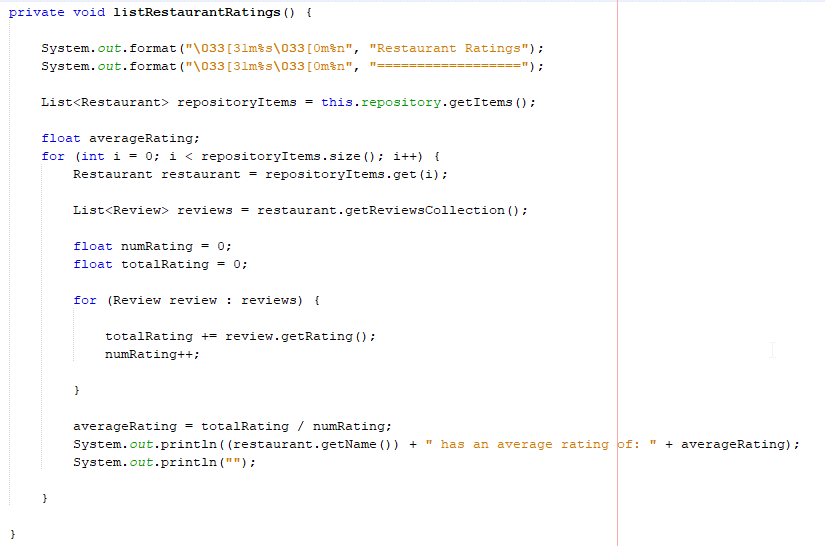
Next a new list of restaurants is created called newListRestaurant, and then the code goes through a for loop which is the size of the repositoryItems (if repository size is 5, then it loops 6 times). This loop gets the restaurant at the current i index, and stores this to a new Restaurant object called getRestaurant. Then the location of the restaurant of the restaurant is returned by getLocation, and saved to a String called restaurantloc.

A nested if loop is run to check the equality of the input location variable that was created by the users input and the location of the current object. If the input variable and current restaurants location is the exact same, then the current restaurant is added to the previously new list of restaurant that was created – called newListRestaurants.

A new set of restaurants is created called newSet, and this is a TreeSet which uses the RestaurantNameComparator method, and this newListRestaurant is added to that newSet and this sorts the restaurants in that list by name order, and the final line outputs those restaurants in the new order.



**Increment 5**

This method is to list the average rating of each restaurant.

The first thing that is done is to create a list of restaurants called repositoryItems and store whats in the current repository object to that variable.

A float variable is created to store the final average ratings of the restaurant in question.

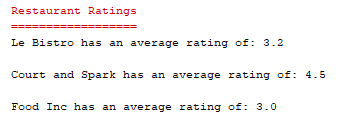
A for loop is ran which loops through until the end of the repositoryItems.

The current restaurant of index i in repositoryItems is read into a new restaurant object called ‘restaurant’, and then the list of reviews of that restaurant is read into a new list of reviews called ‘reviews’

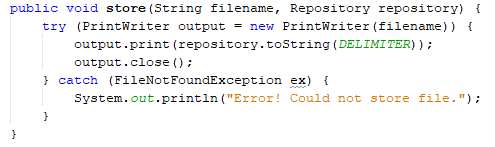
Two float variables two hold he number of ratings and the total ratings are created and set to zero.

And for each ‘review’ in the list of reviews, the current rating of that review in question is added onto the total rating and then numRating is incremented by 1 to keep track of the amount of reviews that have been looped through to calculate the average.

Outside of that nested for loop the average is calculated by dividing the number of reviews by the total rating, for that restaurant object in question. Then in the print statement the name of the restaurant is retrieved by getName, and then added to that output is the average rating. This loops through for each restaurant that exists.



**Persist restaurant collection to a file**

This method is called when the user exits the program. When existing the program, the user is asked to give a name of the file which is passed into this store method, and the current repository object that was stored in memory is also passed into this method.

The method then calls the toString method of the repository, which formats the repository object to the correct format. If the file couldn’t be stored then an error is shown,

