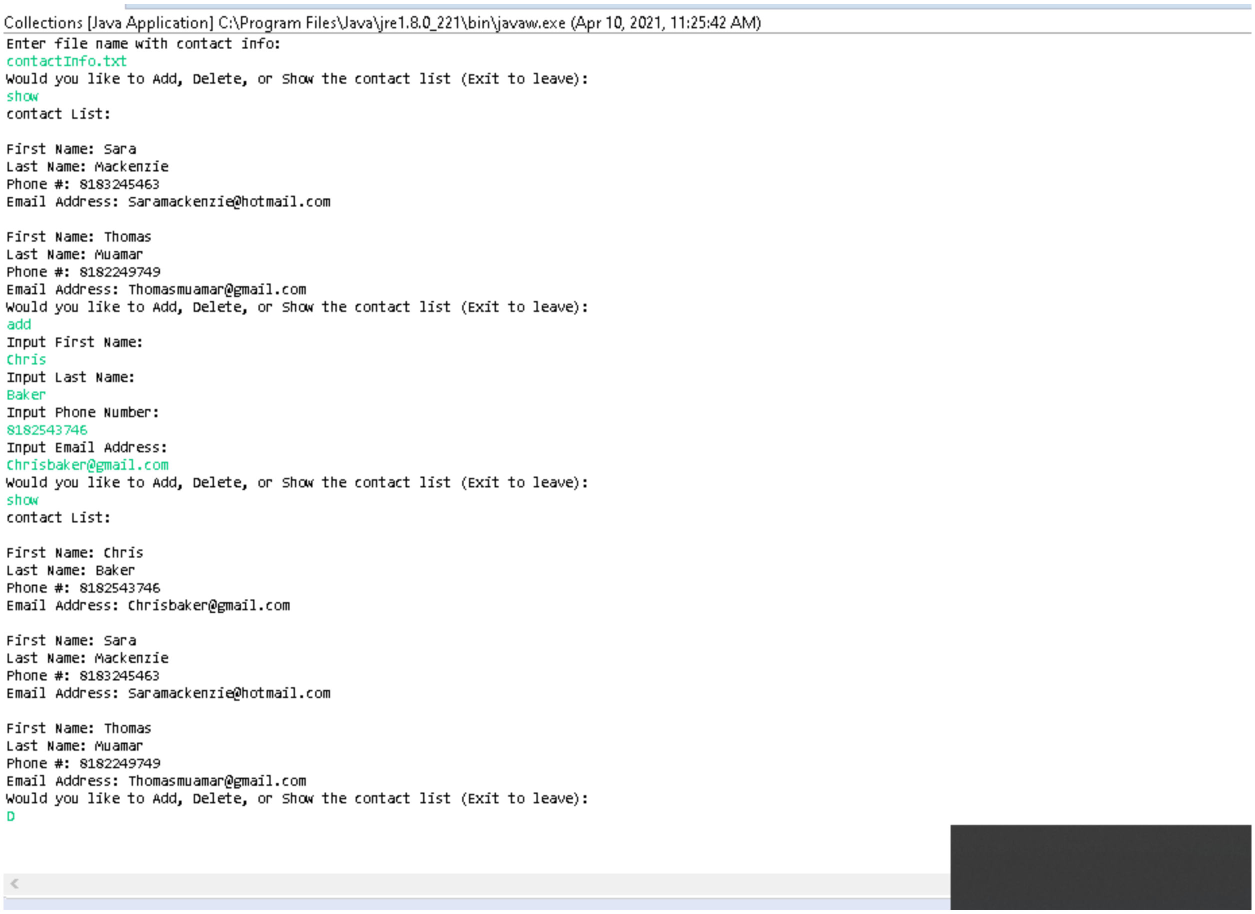
This program allows for user to read a file with contact information and then allows them to change information within this file. In the collection class this program starts with initializing variables and making a scanner sequence to allow user to input the file name. Once user user inputs the file name it is then read through a java character stream of bufferedreader and filereader. Then we take the file information and split and trim it so the information can come out correctly when displaying the file. The do/while loop was used with a if/else statement to allow for user input of adding, removing, and displaying the information. Using equalsIgnoreCase allows for user to not have to capitalizing when inputting what they would like to do . When user decides to add information they input all the information required and then I use the method addingOnList to add this information by using the TreeMap method put(). Then in the remove the user inputs the last name of the individual they would like to remove from the list and it goes through the method removeFromList which uses the TreeMap method remove(). Lastly when individual types show it will display the contact information using the for loop to list the whole entire information will then display in alphabetic order according to last name. Didn’t need a comparator because I took advantage of the sorting it does with keys. the information class just uses a bunch of setters and getters to allow for the format of how the information will be printed out with a stringtoString method. Each of the getters returns the value.

```java
/**
 * This class takes the information in the file and allows to add, delete
 *, and remove from that file of information.
 *@author Thomas Muamar
 */
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;
import java.util.TreeMap;

public class Collections
{
   // intializing variables
   private static Scanner sc = new Scanner(System.in);
   static String contactInfo;
   private static BufferedReader file;
   public static void main(String[]args) throws IOException
   {
     TreeMap<String, Information> contact = new TreeMap<String, Information>();
     if(args.length != 1)
     {
        // allows user to input text file name
     System.out.println("Enter file name with contact info: ");
     contactInfo = sc.nextLine();
     // java character stream to read the file inputed
     file = new BufferedReader(new FileReader(contactInfo));
     String line = "";

     while((line = file.readLine()) !=null)
     {
        // allows for the information to be displayed in a correct format
        String split[] = line.split(" ");
        contact.put(split[1].trim(), new
Information(split[0].trim(),split[1].trim(),split[2].trim(),split[3].trim()));
     }
     // do while loop so when user finishes with one choice they can do it again
     do
     {
        System.out.println("Would you like to Add, Delete, or Show the contact list
(Exit to leave): ");
        String input = sc.nextLine();
        if(input.equalsIgnoreCase("Add"))// after user inputs information uses adding
on list method
        {
           System.out.println("Input First Name: ");
           String firstName = sc.nextLine();

           System.out.println("Input Last Name: ");
           String lastName = sc.nextLine();

           System.out.println("Input Phone Number: ");
           String phoneNumber = sc.nextLine();

           System.out.println("Input Email Address: ");
           String emailAddress = sc.nextLine();
           addingOnList(contact, new Information(firstName, lastName, phoneNumber,
emailAddress));
        } else if(input.equalsIgnoreCase("Delete")) // user inputs lastname that uses
the removefromlist method
        {
           System.out.println("Input Last Name of person to Delete: ");
           String lastName = sc.nextLine();
           removeFromList(contact, lastName);
        }else if (input.equalsIgnoreCase("Show"))
        {
           System.out.println("contact List: ");
           for (String list : contact.keySet())
           {
              System.out.println(contact.get(list));
           }
        }else
        {
           System.exit(0);
        }

     }while(true);
   }
   }
   /**
    * This removes the individual from the tree map by the last name that is
    * inputed by the user.
    * @param contact tree map that stores all the information
    * @param lastName removal of individual by there last name.
    */
   private static void removeFromList(TreeMap<String, Information> contact, String
lastName)
   {
      contact.remove(lastName);
   }
   /**
    * adds the information to the list from the information provided in the
    * if/else statement above.
    * @param contact tree map that stores all the information
    * @param addNew uses the information class to add the infomartion to the file.
    */
   public static void addingOnList(TreeMap<String, Information>contact, Information
addNew)
     {
        contact.put(addNew.getLastName(), addNew);
     }

}

/**
 * this will allow for the information to be outputed by the file
 * in the format indicated in the stringToString method.
 * @author thomas muamar
 *
 */
public class Information
{
   // intializing variables
   private String firstName;
   private String lastName;
   private String phoneNumber;
   private String emailAddress;

   public Information(String fName, String lName, String num, String email)
   {
      firstName = fName;
      lastName = lName;
      phoneNumber = num;
      emailAddress = email;
   }
   // bunch of setters and getters to return the values inputted by user
   public String getFirstName()
   {
      return firstName;
   }
   public void setFirstName(String firstName)
   {
      this.firstName = firstName;

   }

   public String getLastName()
   {
      return lastName;
   }
   public void setLastName(String lastName)
   {
      this.lastName = lastName;
   }
   public String getEmailAddress()
   {
      return emailAddress;
   }
   public void setEmailAddress(String emailAddress)
   {
      this.emailAddress = emailAddress;
   }
   public String getPhoneNumber()
   {
      return phoneNumber;
   }
   public void setPhoneNumber(String phoneNumber)
   {
      this.phoneNumber = phoneNumber;
   }
   // returns the values inputed by the user in this set format shown below
   public String toString()
   {
      return "\nFirst Name: "+firstName+ "\nLast Name: " +lastName+ "\nPhone #:
"+phoneNumber+ "\nEmail Address: " +emailAddress;
   }
}
```

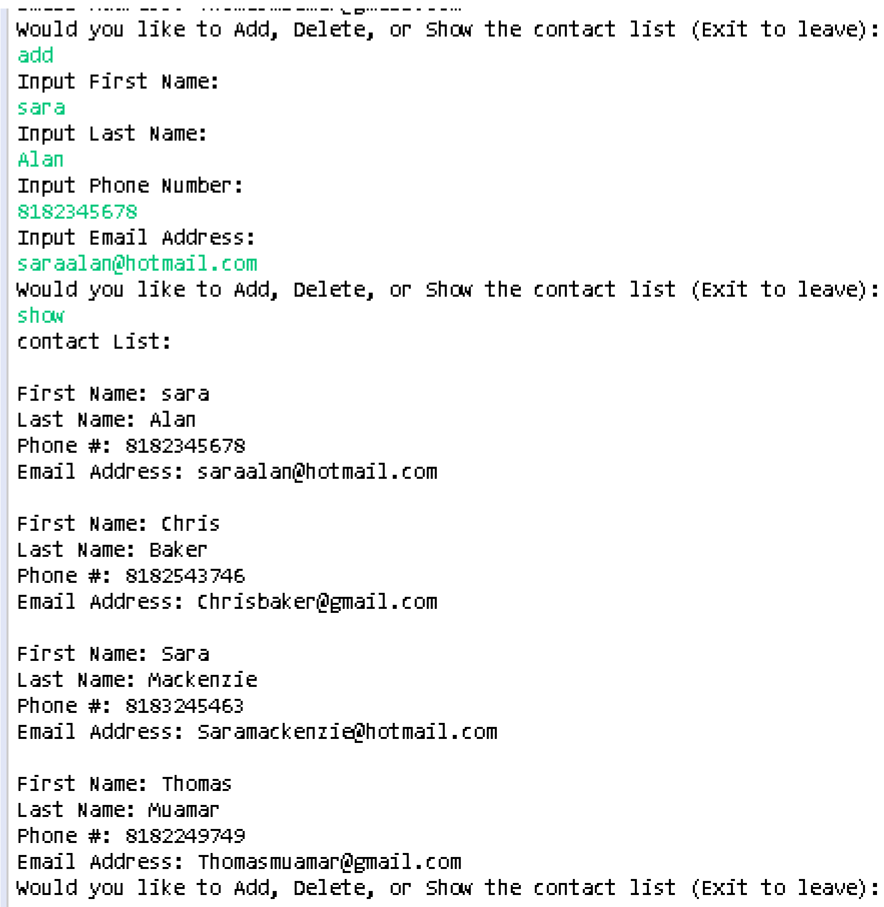
Shows both how adding an individual works and when adding it is organized by last name:



Collections [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (Apr 10, 2021, 11:25:42 AM)

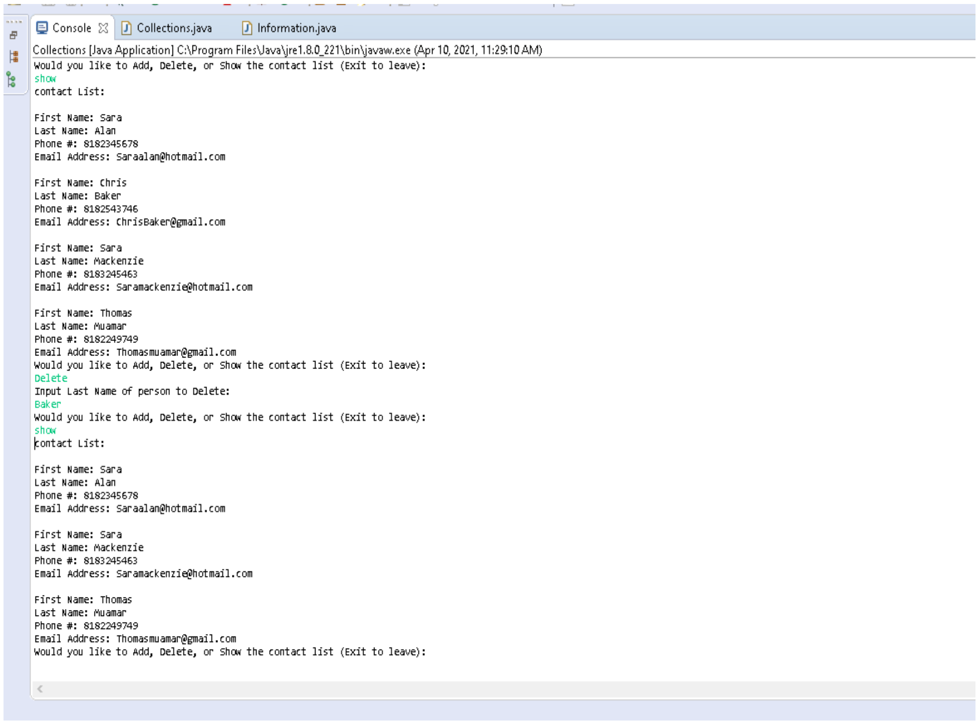
```
Enter file name with contact info:
contactInfo.txt
Would you like to Add, Delete, or Show the contact list (Exit to leave):
show
contact List:

First Name: Sara
Last Name: Mackenzie
Phone #: 8183245463
Email Address: Saramackenzie@hotmail.com

First Name: Thomas
Last Name: Muamar
Phone #: 8182249749
Email Address: Thomasmuamar@gmail.com
Would you like to Add, Delete, or Show the contact list (Exit to leave):
add
Input First Name:
Chris
Input Last Name:
Baker
Input Phone Number:
8182543746
Input Email Address:
Chrisbaker@gmail.com
Would you like to Add, Delete, or Show the contact list (Exit to leave):
show
contact List:

First Name: Chris
Last Name: Baker
Phone #: 8182543746
Email Address: Chrisbaker@gmail.com

First Name: Sara
Last Name: Mackenzie
Phone #: 8183245463
Email Address: Saramackenzie@hotmail.com

First Name: Thomas
Last Name: Muamar
Phone #: 8182249749
Email Address: Thomasmuamar@gmail.com
Would you like to Add, Delete, or Show the contact list (Exit to leave):
D
```

[redacted]



```
Email Address: Thomasmuamar@gmail.com
Would you like to Add, Delete, or Show the contact list (Exit to leave):
add
Input First Name:
sara
Input Last Name:
Alan
Input Phone Number:
8182345678
Input Email Address:
saraalan@hotmail.com
Would you like to Add, Delete, or Show the contact list (Exit to leave):
show
contact List:

First Name: sara
Last Name: Alan
Phone #: 8182345678
Email Address: saraalan@hotmail.com

First Name: Chris
Last Name: Baker
Phone #: 8182543746
Email Address: Chrisbaker@gmail.com

First Name: Sara
Last Name: Mackenzie
Phone #: 8183245463
Email Address: Saramackenzie@hotmail.com

First Name: Thomas
Last Name: Muamar
Phone #: 8182249749
Email Address: Thomasmuamar@gmail.com
Would you like to Add, Delete, or Show the contact list (Exit to leave):
```

Removing an individual:



```
Collections [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (Apr 10, 2021, 11:29:10 AM)
Would you like to Add, Delete, or Show the contact list (Exit to leave):
show
contact List:

First Name: Sara
Last Name: Alan
Phone #: 8182345678
Email Address: Saraalan@hotmail.com

First Name: Chris
Last Name: Baker
Phone #: 8182543746
Email Address: ChrisBaker@gmail.com

First Name: Sara
Last Name: Mackenzie
Phone #: 8183245463
Email Address: Saramackenzie@hotmail.com

First Name: Thomas
Last Name: Muamar
Phone #: 8182249749
Email Address: Thomasmuamar@gmail.com
Would you like to Add, Delete, or Show the contact list (Exit to leave):
Delete
Input Last Name of person to Delete:
Baker
Would you like to Add, Delete, or Show the contact list (Exit to leave):
show
contact List:

First Name: Sara
Last Name: Alan
Phone #: 8182345678
Email Address: Saraalan@hotmail.com

First Name: Sara
Last Name: Mackenzie
Phone #: 8183245463
Email Address: Saramackenzie@hotmail.com

First Name: Thomas
Last Name: Muamar
Phone #: 8182249749
Email Address: Thomasmuamar@gmail.com
Would you like to Add, Delete, or Show the contact list (Exit to leave):
```