Create a detailed design-level description of the structure of your program.

This includes:

* A list of the classes that you expect to use in your project. Some of these classes should capture the basic objects that exist in the problem. There may also be classes that are for the graphical user-interface, or the client (i.e. main method). This list should include the classes of all ADTs that you plan to use in your project, including those provided in the textbook or by the Java API

Some classes I’m thinking about using are:

1. String - to hold various information about
2. Date/Calendar - to set when the contact was created and the birthday of the contacts
3. SimpleDateFormat - used for formatting the dates into strings and mm/dd/yy format
4. Scanner - used for input from the user
5. ArrayList (to store the contacts)
6. Contact (self created class) - used to store the info of the contacts
7. textColors (interface found online to help me change the color of the font in println)
8. Comparable<T> interface (in this case Comparable<Contact> since I’m comparing the contacts in the list to the others to sort the arraylist)
9. Character - to check if the first letter of the first and last name are uppercase
10. Objects - to check if inputs are null or not
11. StringBuffer - to have a flexible string to add to and remove stuff from
12. IllegalLengthException - self created exception class to throw when there is an incorrect length of something

* For each class that you defining on your own, you should specify its interface including all of it public methods. Additionally, include a description of the purpose of the class that will implement this interface and the specification of its data attributes and constructors. You may use a UML class diagram to do this.

The class that will implement the contactInterface will be a class called “Contact”. The class will contain all of the methods mentioned in the interface. It will hold information about each contact like the first and last name (both of type String) and if provided, the birthday (tpye date), phone number (type String), address (type String), and any notes about the contact (type StringBuffer, to be able to potentially hold a huge amount of info). It will also have a variable to see if the contact is empty. If nothing is set in the variable than it will be false, otherwise it will be true.

There will be a getter and setter methods for each of these variables, there will be a isEmpty method call for the isEmpty variable.

With the notes variable, there will be two methods to set the variable. One will be to set the notes and one to add to it.

There will also be 2 setter methods for adding a birthday, one with one date as a parameter and one with 3 ints as parameters, one for the month, day, and year.

There will also be a variable of type date to say when the contact was created.

There will be a method to convert the first letter of the first and last names if they aren’t capitalized.

There will also be a method to add the first and last name all in one go.

Another method was made to see if the given year was a leap year and another method to return a string that set the first variable to uppercase.

There will be a method to check is the input was null, it throws an exception with the provide message if the object given is null.

A toString method was created to display the information about the contact.

The constructors will have different parameters. There will be a default constructor, one with first and last name as parameters, another with those plus the address, and the others will have the previous parameters plus one more of the variables contained in the class, with the exception of the creation time.

I will have another interface that I found online to store all the string variables that help me change the color of the font of the strings being printed using println.

In the main class, there will be a method to handle each of the main choices (add, remove, edit, printContacts, range).

Add method will let the user create and add a contact into the list.

Another method called addContact accept a contact parameter and actually add it to the list.

Edit will edit a contact, remove will remove a specific contact, and range will ask the user for what letter of the last names they want to get the range in using the print in range method which accepts 2 characters as parameters.

There will be another method to ask the user what singel contact they want to print out and it prints it out after accepting the first and last name.

There will be another method to print the contacts as a list and only their names in a numbered list.

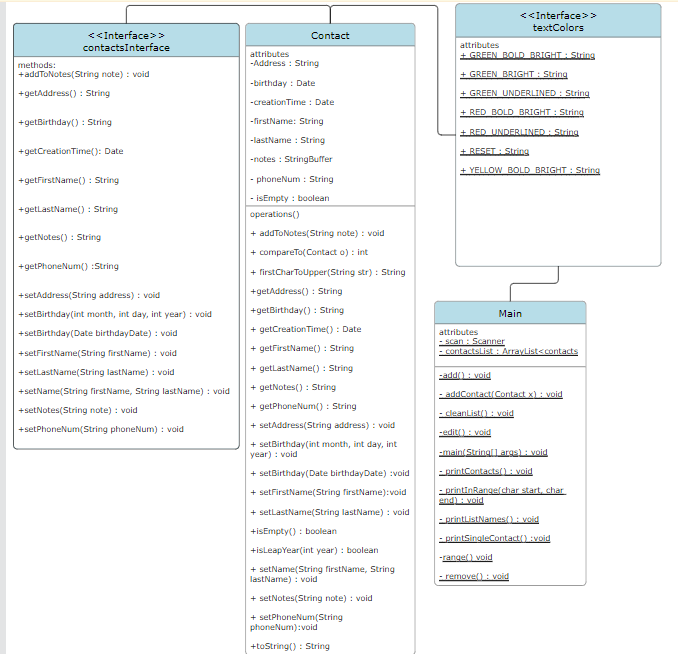
There will be a method to “clean” the list, ie, get rid of any empty contacts.

How Everything comes/works together

<https://app.smartdraw.com/?nsu=1> (uml diagram creator)

<https://app.smartdraw.com/editor.aspx?templateId=d1553a61-d98e-4aa1-a5bc-11a11876ea36&flags=128#depoId=50971848&credID=-55284226>

See next page for uml diagram



Updated version ^^^ (older version below)

