Lab # 6a – Writing and Testing an ArrayList Class

GOALS:

- Practice using Jar files
- Gain a better understanding of how the ArrayList class is implemented
- Practice testing methods within the ArrayList class

Problem:

The purpose of this lab is to develop and test the ArrayList class. I have provided a partial implementation of ArrayList class, along with documentation for each of the methods defined in ArrayList. You will write and test each method/constructor prior to writing the next one.

Creating the Project:

- Create a new Java project called ***csc130.yourname.lab6a***.
- Copy **sqlite-jdbc-3.39.3.0.jar (or download a newer version)** to the root of the project folder and add it as a library to the Java Build Path for the project. This jar file is required for the getData method in Lap6aApp.
- Copy the **BestBuyProducts.db** file to the project folder. This is an SQLite database file containing products.
- Copy the Java files into the project. Store the database file and the jar file at the root of the project.

Exercise

In the exercises below you are asked to include code in the application class in order to test the methods defined in the ArrayList class. Leave all of this code so that I can see how each method was tested. Do not modify code that was previously used to test one method in order to test another. **Comments for the methods are provided in List interface; they are intended to provide you with information about what each method is supposed to do.** Make the necessary modifications to each method before beginning to test another. Include appropriate comments in the application class to explain your code.

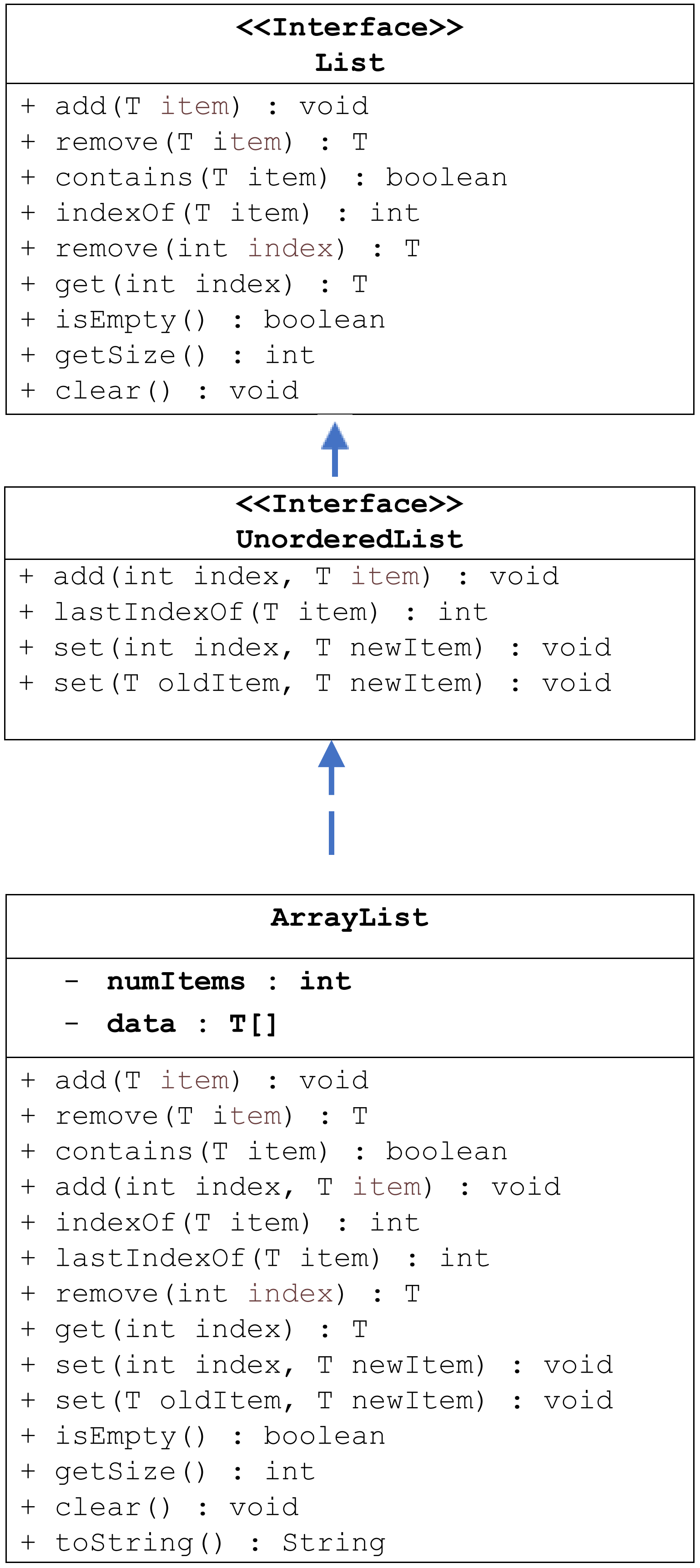
Modify the ArrayList Class:

1. Complete the code to implement all of the methods specified in the ArrayList class.

2. Provide "stubs" (see below) for any methods you haven't yet completed. This will allow you to compile the class and to develop and test each one individually.

   Examples:

```
public void add(T insertItem)      { }
public void  add(int index, T insertItem)  { }
public T remove(T removeItem)             { return null; }
public int indexOf(T searchItem)          { return 0;}
public int lastIndexOf(T searchItem)      { return 0;}
public T set(int index, T newItem)        { return null; }
public T set(T oldItem, T newItem)        { return null; }
```



<<Interface>>
List

```
+ add(T item) : void
+ remove(T item) : T
+ contains(T item) : boolean
+ indexOf(T item) : int
+ remove(int index) : T
+ get(int index) : T
+ isEmpty() : boolean
+ getSize() : int
+ clear() : void
```

<<Interface>>
UnorderedList

```
+ add(int index, T item) : void
+ lastIndexOf(T item) : int
+ set(int index, T newItem) : void
+ set(T oldItem, T newItem) : void
```

ArrayList

```
   - numItems : int
   - data : T[]
```

```
+ add(T item) : void
+ remove(T item) : T
+ contains(T item) : boolean
+ add(int index, T item) : void
+ indexOf(T item) : int
+ lastIndexOf(T item) : int
+ remove(int index) : T
+ get(int index) : T
+ set(int index, T newItem) : void
+ set(T oldItem, T newItem) : void
+ isEmpty() : boolean
+ getSize() : int
+ clear() : void
+ toString() : String
```

Modify Lap6aApp:

3. Define the main method and fix syntax errors as they occur.

4. Create an ArrayList object, **list1**, using the default constructor. Display the contents of the list.

5. Create a second list, **list2**, using the parameterized constructor, and specify a maximum size of 1000. Display the contents of the list.

Modify the ArrayList Class:

6. Provide the necessary code for the ***add*** method, if you have not done it yet.

Modify Lap6aApp:

7. In order to test the ***add*** method you will need to create references to **BestBuyProduct** objects and instantiate them using the parameterized constructor. The data below should be stored in these **BestBuyProduct** objects. Create a variable called ***products*** (a reference to an array of **BestBuyProduct**), call the **getData** method which returns a **BestBuyProduct** array, and store the reference returned in **products**.

   You will have to change the following statement to specify the correct path to the database file:
   `Connection c = DriverManager.getConnection("jdbc:sqlite:`**BestBuyProduct.db**`");`

   Note: if you need to specify a different location (path) for the file, change the path to the BestBuyProduct.db after "`jdbc:sqlite:`.

8. Use the **BestBuyProduct** array from step 7 to insert each of these into **list2**. Create a loop to insert and display the resulting list after each insertion, to determine if the insert method is working. If not, make the necessary corrections and test it again until it works as expected.

Modify the ArrayList Class:

9. Provide the necessary code for the **indexOf** and ***lastIndexOf*** methods.

Modify Lap6aApp:

10. Create a method, **testSearch(key, ulist)**, and write the statement(s) necessary to locate the position of a product in the list. Call the method and pass the product whose id is 1088665 (an item found in the middle of the list). If it is found, display the product id and its location; otherwise display a message stating that the item was not found.

11. Call the ***testSearch*** method and pass the product whose id is 344097 (an item that is not in the list). If it is found, display the product id and its location; otherwise display a message stating that the item was not found.

12. You should test your ***indexOf, lastIndexOf***, and ***contains*** methods with other product ids. Include additional code to search for an item based on index and key field:
    - that is found in the beginning of the list
    - that is found at the end of the list
    - in an empty list (use list1). Be sure to display an appropriate message indicating whether the product was found.

Modify the ArrayList Class:

13. Provide the necessary code for the ***remove*** method.

Modify Lap6aApp:

14. Create a method, **testRemove(key, ulist)**, and write the statement(s) necessary to remove a product from the list. Call the method and pass the product whose id is 1196144 (an item in the middle of the list) and the list. Display the items in the list to ensure it has been deleted.

15. Call the testRemove method and pass the product whose id is 344197 (an item that is not in the list). Display the items in the list to ensure that the list has not changed.

16. You should test your ***remove*** method with other product IDs. Include additional code to remove:
    - an item that is found in the beginning of the list
    - an item that is found at the end of the list
    - a list with 1 item (add a product to list1 and remove it)
    - an item from an empty list (use list1) be sure to display the list after attempting to remove each of these to ensure that your method is working correctly.

17. Comment out the **equals** method in the ***BestBuyProduct*** class. Run your program again. Do you get an error message? Why or why not? How does your output change (if at all)? You should include your answers to these questions in a comment at the bottom of the main method.

18. Remove the comments from the equal method. Run your program again to ensure that you haven't introduced any errors.

19. Create a method in lap6aApp that will display all products containing a string when passed the string and the list. For example: **find("Apple", list2)** should display all products containing the word Apple.

Submit:

Implement all remaining methods in the **ArrayList** class and test them in the Lap6aApp class. Modify/add the comment at the beginning of the files to include an appropriate title, description, and your names as the authors. **Export and submit the entire project to Blackboard.** **Each person must submit the project**.

Sample output:

```
The list contains 0 items.
The list contains 0 items.

Opened database successfully
sku          integer     0
price        float 0
name         text 0
upc          text 0
manufacturer text 0
model        text 0
Records: 1000

The first 5 items in the list:

0 43900,5.489999771118164,"Duracell - AAA Batteries (4-
Pack)",041333424019,"Duracell",MN2400B4Z

1 150115,4.989999771118164,"Energizer - MAX Batteries AA (4-
Pack)",039800011329,"Energizer",E91BP-4

2 324884,39.9900016784668,"Directed Electronics - Viper Audio Glass Break
  Sensor",093207005060,"Directed Electronics",506T

3 346575,16.989999771118164,"Metra - Radio Installation Dash Kit for Most 1989-2000 Ford,
  Lincoln & Mercury Vehicles - Black",086429002757,"Metra",99-5512
4 349572,29.989999771118164,"INSTALL - PORTABLE RADAR DETECTOR
INST",400003495726,"INSTALL",PRT RDR IN

The list contains 1000 items.

Searching for product xxxxxx
Product xxxxxx located at index position: 3

Searching for product xxxxxx
Product xxxxxx was not found in the list

Searching for product xxxxxx
Product xxxxxx located at index position: 0
Searching for product xxxxxx
Product xxxxxx located at index position: 4
Removing product xxxxxx
Product xxxxxx removed from list The list contains 4 items.

Removing product xxxxxx
Product 344d97 not found in the list

Removing product xxxxxx Product
xxxxxx removed from list The
list contains 3 items.  Removing
product 653o09 Product xxxxxx
removed from list
The list contains 2 items.
```