Lab 8: Building the ArrayList Class

Steven K. Andrianoff
Computer Science Department
St. Bonaventure University

Objective:

This lab involves implementing methods of the ArrayList class.

Background:

This lab is based on and extends material in the following section of the textbook:

6.1.3 – *A Simple Array-Based Implementation*

Instructions:

1. Create a Java project in Eclipse. Add the files `ArrayList.java` and `WordLister.java` to your project. `ArrayList.java` is a partially completed generic ArrayList class that we developed in class. `WordLister.java` is a main class that uses the ArrayList class to store a list of words obtained from a text file. Add the data files `catnfiddle.txt`, `jacknjill.txt`, and `pease.txt` to your project. Run the program using one of the text files to be sure everything is set up correctly.

2. Add the *contains* method to the ArrayList class.

Modify the main method to add code to test the contains method. Test the list for a word that is present and for a word that is not present. Run the program, print the results, and hand in the results with your lab.

3. Review the API for ArrayList to understand the *isEmpty* method. Add the isEmpty method to your ArrayList class.

Add code to the main method to test the isEmpty method. Provide the code you used to test the method, print your results, and turn in the code and your results with your lab.

4. Review the API for ArrayList to understand the *set* method. Add the set method to your ArrayList class.

Add code to the main method to test the set method. Provide the code you used to test the method, print your results, and turn in the code and your results with your lab.

5. Review the API for ArrayList to understand the *indexOf* method. Add the indexOf method to your ArrayList class.

Add code to the main method to test the indexOf method. Provide the code you used to test the method, print your results, and turn in the code and your results with your lab.

6. Add the *toString* method to the ArrayList class. Have the toString method behave in the same way as it behaves in the ArrayList class in the Java library.

Modify the main method to use the toString method to display the list that is built. Run the program on `pease.txt`, print the results, and hand in the results with your lab.

7. Add the *equals* method to the ArrayList class. When designing the equals method you need to think about what it means for two lists to be equal. A reasonable expectation is to say two lists are equal if and only if they contain the same data values in the same order. Write the equals method using this idea.

Note: Conceptually the equals method should have the same logic you used in Programming Assignment 2.

Modify the main method to read words from two text files into two different lists. Then display whether or not the two lists are equal. Run the program twice, once in the case where the lists are the same and once in the case where the lists are different. Print the results, and hand in the results with your lab.

Be sure you have properly documented your code then print the files `ArrayList.java` and `WordLister.java` and hand them in with your lab.

Extra Credit:
Add the *remove* method to the ArrayList class. The signature of the method should be:

```
public E remove(int index)
```

Modify the main method by adding code at the end of main to go through the list *wordList* and remove all words that start with 't'. Run the program on the file `catnfiddle.txt`, print the results, and hand in the results with your lab.

Hand in:

The write-up you hand in for this lab should include:

- the results requested in Steps 2-7.
- copies of the source code `ArrayList.java` and `WordLister.java` requested in Step 7.

Help Policy:

Help Policy in Effect for This Assignment: Group Project with Limited Collaboration

In particular, you may discuss the assignment and concepts related to the assignment with the following persons, in addition to an instructor in this course: any member of your group; any St. Bonaventure Computer Science instructor; and any student enrolled in CS 132.

You may use the following materials produced by other students: materials produced by members of your group.

You may use the following materials produced by other students: NONE.