

Java Lab 10: More Memos

This lab will re-use your solution to Lab 8. Create a new project, Lab10. Copy MakeANote into the src package. Make a new package underneath src named NotePackage. Copy all the other classes from Lab 9 into it. IntelliJ should add “package NotePackage” at the top of each .java file there – make sure that’s true.

1. Change the Note class by having it implement the Comparable<Note> interface. Write the compareTo() method that compares notes based on their name data field.

2. Add a method to NoteCollection called sortByName(). It should sort the noteList by calling Collections.sort() with the noteList as a parameter.

3. Add a menu choice to the displayMenu() strings, "Sort by Name", before "Return to previous menu". Add that new case (or if-condition, if you coded it that way) to the display sub-menu processing; don't forget to change the case for "Return to previous menu" and the loop condition. When the user chooses this option, call sortByName(), but don't display it – the user will still have to choose #1 on this submenu.

Test this by adding a few notes with different names; make sure the note list really does display in sorted order.

4. Create a new class (in NotePackage) called NumberSorter that implements the Comparator<Note> interface. Implement its compare() method, comparing two Note objects on their noteNumber data field.

5. Add a method to NoteCollection called SortByNumber(). It should sort the noteList by calling Collections.sort() with noteList as the first parameter and an instance of NumberSorter as the second parameter.

6. Similar to #3: add a menu choice to the displayMenu() strings, "Sort by Number", before "Return to previous menu". Add that new case (or if condition, if you coded it that way) to the display sub-menu processing; don't forget to change the case for "Return to previous menu" and the loop condition. When the user chooses this option, call sortByNumber(), but don't display it – the user will still have to choose #1 on this submenu.

Again, test this by adding a few notes with different names; make sure the note list really does display in sorted order by number; then sort it by name again and display it.

7. One more time: create a new class SizeSorter that implements Comparator<Note>; it should compare the two notes on the size of the String returned by toString(). Add SortBySize() to NoteCollection. Change the display menu one more time. Test by alternating among the three different sorts