Regis University CC&IS
CS310 Data Structures
Programming Assignment 3: Linked Lists

Problem Scenario

The real estate office was very impressed with your work from last week. Nevertheless, the real estate office manager just read his "Java Geek Weekly", and discovered arrays and ArrayLists may not be the most optimal data structures to use. Thus, the real estate office has asked to replace them with Linked Lists.

There was also some concern within the real estate office there were too many errors being allowed into the reports. They would still like the program to **first build the logs** and **run the report** as you did last week, producing a report in the file: `output/assn3initialReport.txt`

But after that, they would like the program to **validate the data in the linked lists,** and **clean them up**. They would like the program to:

- Check the realtor log for invalid realtor license numbers, using the realtor license number validation method already developed for the Realtor class
  - If a realtor license number is invalid, remove the realtor from the realtor log, along with all of the realtor's properties in the property log.
- Check the property log for invalid MLS numbers, using the MLS number validation method already developed for the Property class
  - If a property has an invalid MLS number, remove that property from the list.
- Write lines to the console, as it cleans the lists, indicating any actions taken to remove invalid data.

Once all the data has been validated, the program should **run the final report again, producing a second report**, so the real estate office can compare the two reports.
The second report will be stored in file: `output/assn3cleanReport.txt`

Program Requirements

The program must follow the **CS310 Coding Standards** from Content section 1.9. This **csv** input file for this program will be called "assn3input.txt", and will have the same format as the input file from Assignment 2.

The **implementation classes** from last week will be ***replaced*** with new implementation classes that will implement Linked Lists. So you will modify the implementations from last week, by removing the array and ArrayList data structures and replacing them with this week's Linked List data structures.

The textbook described several types of linked lists, including:

- Singly Linked Lists
- Java Collection Linked List (java.util.LinkedList)

Your **RealtorLogImpl** will implement an ***ordered*** singly linked list, with a Realtor node class, and (add, remove, etc) methods that you code yourself (see Adding to an ordered list in Content 3.5)

Your **PropertyLogImpl** will implement a Java Collection Linked List.

To enable writing reports to different files, you will also need to add a filename ***parameter*** to the **PrintImpl**'s report printing method (and then add arguments to any calls to that method).

©Regis University, All Rights Reserved
Unauthorized distribution (including uploading to any non-Regis Internet website) violates copyright law

Iteration

All of the **RealtorLogImpl** methods that traverse the singly linked list, will start at the top of the list, and traverse the list using a loop.

Since this is an ordered list, when adding new nodes, the code will need to find the correct insertion spot that will maintain order by realtor license numbers.

All of the **PropertyLogImpl** methods that traverse the linked list will use **Iterator**, as described in the textbook.

New Methods

Add a **traverseDisplay()** method to both **RealtorLogImpl** and **PropertyLogImpl**.
These methods will first display a header:

```
Realtor Log:          OR    Property Log:
```

and will traverse the list being implemented, using the **toString()** method to display each object in the list.

Add a **cleanUp()** method to the **RealtorLogImpl** to validate and clean up the realtor list, removing Realtor objects with invalid realtor license numbers.

Add a **cleanUp()** method to the **PropertyLogImpl** to validate and clean up the property list, removing Property objects with invalid MLS numbers.

Node Class

The textbook refers to inner classes – you have probably not seen these yet. Inner classes allow you to include one class within another. Using the inner class, textbook examples accessed attributes like this

Inserting a Node in a List

If we have a reference `harry` to node `"Harry"`, we can insert a new node, `"Bob"`, into the list after `"Harry"` as follows:

```
Node<String> bob = new Node<String>("Bob");
bob.next = harry.next; // Step 1
harry.next = bob;      // Step 2
```

The linked list now is as shown in Figure 2.17. We show the number of the step that created each link alongside it.

Since all of the attributes were private, and within the parent class, it was possible to use the "dot" (.) notation to access the attributes.

You will ***not*** do this. Instead of using inner classes, you will be creating your own **Node** class for the Realtor's singly linked list. Since the **Node** class will be a separate class, you will use getter/setter methods for access to private attributes. For example, using a **Node** class, the above code would be:

```
Node<String> bob = new Node<String>("Bob");
bob.setNext(harry.getNext());     // Step 1
harry.setNext(bob);               // Step 2
```

A sample **Node** class for a singly linked list is shown in the linked list building example of the online Content in section 3.2.

The main method

You will start with the same **main** method that you used for Assn 2.

©Regis University, All Rights Reserved
Unauthorized distribution (including uploading to any non-Regis Internet website) violates copyright law

After reading the data and creating the initial report (same as in Assn 2), add code to the end of the **main** method to:

- Display each list, using the **traverseDisplay()** methods.
- Clean up the lists.
- Create the second "clean" report.

Hints

Your application should contain the following class source code files now:

- CS310<lastname>.java
- RealtorNode – node class for a singly linked list
- RealtorLogImpl – singly linked list implementation of the realtor log
- PropertyLogImpl – **java.util.LinkedList** implementation of the property log
- PrintImpl – same format and data displayed as last week, but modified to use the linked lists

Additional Requirements

- Your input data file will still be read from the **input** folder in your project.

  Place all test data files that you create to test your program in the **input** folder of your project, and name them as follows:

  `assn3input1.txt`
  `assn3input2.txt`
  (i.e. number each data file after the filename of **`assn3input.txt`**)

  Together, all of your test data files should demonstrate that you have tested every possible execution path within your code, including erroneous data which causes errors or exceptions.

  For example, make sure you have test files with realtor license numbers that are **not** in the correct order, so you can validate that your program is inserting them in order.

  WARNING: You will not be given ANY credit for the input file that is included as an example with these requirements. The test files should be of your own creation.

- Your output data files will still be written to the **output** folder in your project.
- Create and/or modify **Javadoc headers**, and generate **Javadoc files**
- Add screen shots of **clean compile** of your classes to the documentation folder.

  WARNING: Submittals without the clean compile screenshots will **not** be accepted.
  (This means that programs that do not compile will **not** be accepted)

Program Submission

This assignment is due by midnight of the date listed on the **Course Assignments by Week** page.

- Export your project from NetBeans, following the same steps used in Assns 1 &2.
  - Name your export file in the following format:
    `CS310<lastname>Assn<x>.zip`

    For example: **`CS310SmithAssn3.zip`**

    NOTE: Save this zip file to some other directory, not your project directory.

©Regis University, All Rights Reserved
Unauthorized distribution (including uploading to any non-Regis Internet website) violates copyright law

- Submit your **.zip** file to the **Prog Assn 3** Submission Folder (located under **Assignments** tab in online course).

  Warning: Only NetBeans export files will be accepted.
  Do not use any other kind of archive or zip utility.

Grading

This program will be graded using the **rubric** that is linked under **Student Resources** page.

WARNING:

*Programs submitted more than 5 days past the due date will **not** be accepted, and will receive a grade of 0.*

See appendix on next pages for samples of input and output

©Regis University, All Rights Reserved
Unauthorized distribution (including uploading to any non-Regis Internet website) violates copyright law

Appendix – Sample Input/Output

Sample Input Data File:

```
REALTOR,ADD,MN4564567,Carla,Combs,444-555-6666,0.014
PROPERTY,ADD,4455667,MN4564567,4455 This Circle,Denver,CO,80333,1,1,N,344555
REALTOR,ADD,RR6655443,Jerry,Smith,555-444-3333,0.013
PROPERTY,ADD,23456789,RR6655443,888 Terry Lane,Longmont,CO,80503,3,2,N,222222
REALTOR,ADD,AB1234567,Matthew,Munez,123-456-7890,0.012
PROPERTY,ADD,1234567,AB1234567,1234 Which Way,Somewhere,CO,82222,3,3,Y,222222
PROPERTY,ADD,2234567,AB1234567,345 Main St,Fort Collins,CO,81333,4,3.5,N,222333
REALTOR,DEL,MN4564567
REALTOR,ADD,XY98765432,Alex,Yung,999-888-7777,0.013
PROPERTY,ADD,9998888,XY98765432,111 Main St,Cheyenne,WY,82222,1,1,N,199888
```

Sample Display Output:

```
Reading data from file input/assn3Input.txt
  ADDED: Realtor with license MN4564567
  ADDED: Property with MLS number 4455667 listed by realtor MN4564567
  ADDED: Realtor with license RR6655443
    ERROR: Property with MLS number 23456789 has an invalid MLS number.
  ADDED: Property with MLS number 23456789 listed by realtor RR6655443, regardless of data errors.
  ADDED: Realtor with license AB1234567
  ADDED: Property with MLS number 1234567 listed by realtor AB1234567
  ADDED: Property with MLS number 2234567 listed by realtor AB1234567
  DELETED: Realtor with license MN4564567 has been removed from the realtor log
           All realtor's properties will also be removed from the property log
    Removing property 4455667 listed by Realtor MN4564567
    ERROR: Realtor with license XY98765432 has invalid license number.
  ADDED: Realtor with license XY98765432, regardless of data errors.
  ADDED: Property with MLS number 9998888 listed by realtor XY98765432
Done reading file. 10 lines read

Realtor Log:
    Realtor{licenseNum=AB1234567, firstName=Matthew, lastName=Munez, phoneNum=123-456-7890, commissionRate=0.012}
    Realtor{licenseNum=RR6655443, firstName=Jerry, lastName=Smith, phoneNum=555-444-3333, commissionRate=0.013}
    Realtor{licenseNum=XY98765432, firstName=Alex, lastName=Yung, phoneNum=999-888-7777, commissionRate=0.013}
Property List:
    Listing{mlsNum=23456789, licenseNum=RR6655443, streetAddress=888 Terry Lane, city=Longmont, state=CO, zipCode=80503, numBedrms=3,
numBaths=2.0, sold=false, askingPrice=222222.0}
    Listing{mlsNum=1234567, licenseNum=AB1234567, streetAddress=1234 Which Way, city=Somewhere, state=CO, zipCode=82222, numBedrms=3,
numBaths=3.0, sold=true, askingPrice=222222.0}
    Listing{mlsNum=2234567, licenseNum=AB1234567, streetAddress=345 Main St, city=Fort Collins, state=CO, zipCode=81333, numBedrms=4,
numBaths=3.5, sold=false, askingPrice=222333.0}
    Listing{mlsNum=9998888, licenseNum=XY98765432, streetAddress=111 Main St, city=Cheyenne, state=WY, zipCode=82222, numBedrms=1,
```

©Regis University, All Rights Reserved
Unauthorized distribution (including uploading to any non-Regis Internet website) violates copyright law

```
numBaths=1.0, sold=false, askingPrice=199888.0}

Creating initial report...
Report is complete -- located in file: output/assn3initialReport.txt

Cleaning up realtor and property logs...
Invalid realtor license number for realtor XY98765432
    Removing realtor XY98765432 and all propertys associated with this realtor
    Removing property 9998888 listed by Realtor XY98765432
Invalid MLS number for property 23456789 -- Deleting property from log

Creating clean report...
Report is complete -- located in file: output/assn3cleanReport.txt
```

Sample Initial Report:

```
AB1234567  Munez, Matthew

        1234567      1234 Which Way    3/3.0  $  222222.00  SOLD
                Somewhere, CO 82222

        2234567         345 Main St    4/3.5  $  222333.00
             Fort Collins, CO 81333

    Number of Property Listings for Realtor: 2
    Total sales value of Property Listings for Realtor: $ 444555.00

RR6655443  Smith, Jerry

       23456789      888 Terry Lane    3/2.0  $  222222.00
                 Longmont, CO 80503

    Number of Property Listings for Realtor: 1
    Total sales value of Property Listings for Realtor: $ 222222.00

XY98765432  Yung, Alex

        9998888         111 Main St    1/1.0  $  199888.00
                 Cheyenne, WY 82222

    Number of Property Listings for Realtor: 1
    Total sales value of Property Listings for Realtor: $ 199888.00

Total Number of Property Listings for ALL Realtors = 4
Total sales value of Property Listings for ALL Realtors = $ 866665.00
```

©Regis University, All Rights Reserved
Unauthorized distribution (including uploading to any non-Regis Internet website) violates copyright law

Sample Clean Report:

```
AB1234567  Munez, Matthew

        1234567      1234 Which Way    3/3.0  $  222222.00  SOLD
                Somewhere, CO 82222

        2234567         345 Main St    4/3.5  $  222333.00
             Fort Collins, CO 81333

    Number of Property Listings for Realtor: 2
    Total sales value of Property Listings for Realtor: $ 444555.00

RR6655443  Smith, Jerry

    Number of Property Listings for Realtor: 0
    Total sales value of Property Listings for Realtor: $ 0.00

Total Number of Property Listings for ALL Realtors = 2
Total sales value of Property Listings for ALL Realtors = $ 444555.00
```

©Regis University, All Rights Reserved
Unauthorized distribution (including uploading to any non-Regis Internet website) violates copyright law