**Instructions**

This assignment is functionality-wise exactly the same as assignment A02-RealEstate, however you will be using a linked-list and recursion in the back-end. Watch out for the following

* Instead of using an array you will use a linked list of houses created by you from scratch. This list will be called HList.java.
  + HList is a linked list of HNodes which will contain a House object.
* When printing the list you will do it using **recursion**.

**Instructions:**

Using the**HouseException.java** and the **House.java** from A01-Houseyou will implement a driver class called **HouseListing.java** to manipulate a Linked list (HList) of Home objects. This time, again,  you will be creating the driver class.

**For this assignment you will learn the following:**

* Using linked lists to manipulate objects
* using recursion to print linked lists

**HNode.java- node containing a House object.** *(total of  120 points distributed as shown below)*  Note that the method UML description is listed below and it should be implemented exactly as it is.

* +HNode( ) - constructor, takes no parameters. *(20 points)*
* +setNext(HNode n): void *(20 points)*
* +getNext( ): HNode  *(20 points)*
* +setHouse(House h): void  *(20 points)*
* +getHouse( ): House  *(20 points)*
* +toString( ): String  *(20 points)*

You do not need a Node exception because the House is already validated with the HouseException. However, you may write it and submit it, if your design requires it. Please comment on your design.

**HList.java  -list of HNodes - list of nodes containing House objects***(270  points)* This class works directly with the driver class to implement each one of the menu options and to create the list object.

* +HList( )   constructor, takes no parameters, initializes list, instance variables and references *(20 points)*
* +add(House): void  adds a HNode to the list. ***You may choose to send an HNode or a House as a parameter***. It depends on how you design your code. Just add a comment about your design choices. This method will be called to implement the add house option from the menu. *(75 points)*
* +remove(int MLS): boolean  removes the house with the given MLS, if it exists. This method will be called to implement the remove house option from the menu. *(75 points)*
* +printAllHouses( ):void prints all the houses in the list. This method is called to implement option 4. *Must be a****recursive****method (50 points,* If you create a non-recursive method you will only get  25 points).
* +printHousesLessThan(double price):void prints the houses priced less than the parameter given.It implements option 3.  *Must be a****recursive****method. (50 points,* If you create a non-recursive method you will only get  25 points).

**Driver class HouseListing.java**

**When your code begins execution the following menu should be displayed. It is the same menu and requirements as in the previous RealEstate.java assignment**

1. add a house  
2. remove a house  
3. print houses that cost less than a given price  
4. print all the houses  
0. end this program

**Your driver class should:**

* be named: **HouseListing.java**
* work with your Home.java and HomeException.java (**fix them** if you did not get full credit, otherwise you will lose even more points this time)
* validate user input. Your code should not crash.
* create an HList object to manipulate Nodes and House objects

**Implementing the menu options**

1. add a house (call the add method)  
2. remove a house (call the remove method)  
3. print houses that cost less than a given price (call the printHousesLessThan method)  
4. print all the houses (call the printAllHouses method)  
0. end this program

**Implementing the looping menu in the driver class:**

* Your code should continue to loop until the user selects 0. If the program terminates at any other point for any other reason you will lose these points.  *(10 points)*.
* If the user enters an out of range number, you should print a carefully crafted error message explaining the problem and loop into the menu again. *(10 points)*.
* If the user enters anything other than a number, you should catch the exception, print a carefully crafted error message explaining the problem and loop into the menu again. *(10 points)*.

**Your code should always continue to loop until the user selects 0.**

**Other considerations**

* The linked-list should not contain duplicate house MLSs at any time.
* You should not use any of the linked-list methods implemented by Java. Everything should be done from scratch by you.
  + If you need to import anything to make your linked list work then you are doing it the wrong way. Follow the lecture!
  + If you have any questions please send me an email.
* You should use the proper exceptions, like InputMismatchException or HouseException,  NullPointerException etc as appropriate
* You may use only ONE linked list for this assignment. This should be a linked list of HNodes containing House objects.
* You may implement extra methods to get this assignment done, however you MUST have the required methods to get the points listed.

When you submit your assignment, you **MUST**submit the following:

* HouseListing.java  <----- This must be the driver class.
* HList.java
* HNode.java
* House.java (must resubmit) *(10 points, if incorrect will lose 20points)*
* HouseException.java (must resubmit)  *(10 points, if incorrect will lose 20points)*

any other classes you implemented for this assignment must be submitted as well.

[**You must Follow the Java coding standards**](http://www2.hawaii.edu/~tp_200/ics111/material/codingStandards.html)and put your name at the top of your code. *(20 points)*