Homework #2
Linked List

Submission guideline:

Submit the homework through blackboard. Before submission make sure your codes do not have any error, unexecutable code and/or late submission will not receive any credit. Submit the code (.java files only) as a single .zip archive (right click on the folder which you want to zip and select compress). You can include a README text file to give the TAs instructions about your submission. The .zip file name should be in the following format:

<firstname>_<lastname>_<id>_hw<num>.zip

For example, if John Doe with student ID 123456789 is submitting the third homework, the file should be named john_doe_123456789_hw3.zip

Number Distribution:
(1) 5 + 5 + 5 + 5 = 20
(2) 10 + 10 = 20
(3) 10

Total Marks: 50
Homework posted on: Friday, February 16, 2018
Submission Date: Friday, February 23, 2018 (11:59 PM)

Linked List

In this homework, you are given with a basic LinkedList implementation (LinkedList.java) with following methods:

1. size() : returns the size of the list
2. isEmpty() : returns true if the list is empty
3. clear() : removes all nodes from the list
4. get(pos) : returns data at the specified position of the list
5. contains(key) : returns true if the list contains the specified element
6. getFirst() : returns the first element of the list
7. addToFirst(data) : inserts a new element at the beginning of the list
8. removeFirst() : removes the first element of the list
9. getLast() : returns the last element of the list
10. addToLast(data): adds an element at the end of the list
11. remove(key): removes the first occurrence of the specified element in the list

Also, there is a LinkedListTest class given for testing.

Problem 1

Enhance the LinkedList class by adding following methods to do some more operations of the linked list:

1. insertAt(pos, data): inserts the element data at position pos of the list. If position exceeds the length of the list throw an exception.

   Example:

   List: 4 6 7 2
   insertAt(2, 8)
   List: 4 6 8 7 2
   insertAt(0, 9)
   List: 9 4 6 8 7 2
   insertAt(-1, 10)
   IllegalArgumentException
   List: 9 4 6 8 7 2
   insertAt(8, 12)
   IndexOutOfBoundsException
   List: 9 4 6 8 7 2

2. removeAt(pos): remove element at position pos. If position exceeds the length of the list throw an exception.

   Example:

   List: 4 6 7 2 5 9
   removeAt(2)
   List: 4 6 2 5 9
   removeAt(0)
   List: 6 2 5 9

removeAt(-1)
IllegalArgumentException
List: 6 2 5 9
removeAt(7)
IndexOutOfBoundsException
List: 6 2 5 9

3. insertAfter(key, data): insert a new element data after the first occurrence of key.
   Example:
   List: 4 6 7 2 5 9
   insertAfter(2, 3)
   List: 4 6 7 2 3 5 9
   insertAfter(1, 8)
   IndexOutOfBoundsException
   List: 4 6 7 2 3 5 9

4. insertBefore(key, data): insert a new element data before the first occurrence of key.
   Example:
   List: 4 6 7 2 5 9
   insertBefore(2, 3)
   List: 4 6 7 3 2 5 9
   insertBefore(1, 8)
   IndexOutOfBoundsException
   List: 4 6 7 3 2 5 9

Problem 2

Implement sorting and duplicate removal operations on the linked list:

1. sortList(): sort the list in ascending order of the keys
   Example:
   List: 4 6 7 2 5 9
   sortList()
   List: 2 4 5 6 7 9

2. removeDuplicates(): remove duplicate elements from the list.
   Example:
   List: 2 4 4 1 1 3 9
   removeDuplicates()
   List: 2 4 1 3 9

Problem 3

Implement a mergeLists method in LinkedListTest class that merges 2 given lists and returns an output list. While merging make sure:

1. Output list is sorted
2. Remove any duplicate

Example:

list1: 2 4 7 9

list2: 4 5 1 6 7

merge (list1, list2)

Output: 1 2 4 5 6 7 9

Testing

Input:
In the input files, the first line contains the elements of a linked list. Subsequent lines contain the commands for different operations on the linked list. Commands are encoded as follows:

Function Name	Command Code	Input Format
insertAt	1	1 <param1> <param2>
removeAt	2	2 <param1>
insertAfter	3	3 <param1> <param2>
insertBefore	4	4 <param1> <param2>
sortList	5	5
removeDuplicates	6	6

There are 2 input files:
1. input1.txt
2. input2.txt

The LinkedListTest program first creates list1 from the first line of input1.txt and run different commands from the subsequent lines of the file. After that list2 is created and different operations are performed similarly from input2.txt file. Finally, mergeLists() method is called for list1 and list2. After each operation lists are printed in the console.

Example input1.txt

```
4 6 2 8 5
1 3 4
2 4
3 2 3
4 4 2
5
6
```

Example input2.txt

```
7 2 4 6 2
1 3 4
2 4
```

Expected Output

```
reading input file: input1.txt
List: 4 6 2 8 5
insertAt (3, 4)
List: 4 6 2 4 8 5
removeAt (4)
List: 4 6 2 4 5
insertAfter (2, 3)
List: 4 6 2 3 4 5
insertBefore (4, 2)
List: 2 4 6 2 3 4 5
sortList()
List: 2 2 3 4 4 5 6
removeDuplicates()
List: 2 3 4 5 6

reading input file: input2.txt
List: 7 2 4 6 2
insertAt (3, 4)
List: 7 2 4 4 6 2
removeAt (4)
List: 7 2 4 4 2

List 1: 2 3 4 5 6
List 2: 7 2 4 4 2
Output: 2 3 4 5 6 7
```