Assignment LinkedList CSIS-2420

Learning Objectives:

- Deepen your understanding of linked lists
- Implement and test the secified functionality
- Practice developing algorithms

Turning In:

Make sure to include a comment with your name and assignment number on top of each source code file.
Submit an **executable** jar file that **includes your source** code files via Canvas.

Description:

Complete the implementation of class **LinkedList<E>**. Then write a test client that presents the user with a menu to call various methods from the newly created class.

Instructions:

1. This assignment is an extension of the LinkedList that we started already as a lab in class.
   Add the methods specified below (make sure to implement the functionality described in the doc comment)
   Hint: Make sure to test the methods thoroughly – e.g. with an empty list, a key that doesn't exist, etc.

2. Write a test client called LinkedListApp.
   It is the job of LinkedListApp to test the class LinkedList<E> that you just wrote.
   In order to do that create a LinkedList of strings that is initialized with 5 itmes (you choose the theme - e.g. friends, motorcycles, books, coins, etc.)

   The test client should present the user with a **menu** that includes the following options:

```
a .. Size                 // displays the number of elements in the list
b .. Insert               // inserts an element after a user selected key  and element
c .. Remove               // deletes the first node that includes the user selected element
d .. Display all items    // displays all elements like this: [el1, el2, el3, … ]
e .. Clear                // empties the list
q .. Quit
```

   When the user makes a choice provide the requested functionality by calling the methods of class LinkedList<E>.
   Then allow the user to make more choices until s/he selects Quit. (*Hint: you'll need appropriate error handling*)

```java
/**
 *  The size of the list
 *
 * @return The number of elements in the list
 */
public int getSize()
{
   // TODO
}
```

```java
/**
 *  Removes all nodes from the list.
 *
 *  <dt><b>post-condition:</b></dt>
 *  <dd>The list is empty and size is 0</dd>
 */
public void clear()
{
   // TODO
}


/**
 *  Inserts the new element (newElement) after the first node
 *  containing the key.
 *
 *  @param key
 *  The key element after which the new element should be inserted
 *  @param newElement
 *  The new element that needs to be inserted
 *  <dt><b>pre-condition:</b></dt>
 *  <dd>The key is included on the list</dd>
 *  <dt><b>post-condition:</b></dt>
 *  <dd>The newElement has been added immediately after the key</br>
 *  size has been incremented by one</dd>
 *  @throws RuntimeException - If the key was not found in the list
 */
public void insertAfter(T key, T newElement) {
   // TODO
}


/**
*  Removes the first occurrence of the specified key (element)
*  in this list.
*  @param key
*  The element to be removed
*  <dt><b>pre-condition:</b></dt>
*  <dd>The key is included on the list</dd>
*  <dt><b>post-condition:</b></dt>
*  <dd>The key has been removed from the list</br>
*  size has been decremented by one</dd>
*  @throws RuntimeException - If the key was not found in the list
*/
public void remove(T key) {
   // TODO
}
```