

1.

```
package lab19;

import java.util.LinkedList;
import java.util.List;
import java.util.Scanner;

public class insertElementLinkedList {
/**
 * Adds the numbers obtained through the console to the LinkedList until the input
number
 * are 0.
 * @return numbers - the linkedList with all numbers
 */
    public static LinkedList<Integer> addToLinkedList() {
        Scanner kbd = new Scanner(System.in);
        LinkedList<Integer> numbers = new LinkedList<Integer>();
        int number = kbd.nextInt();    //variável afetada ao valor inserido na consola

        while (number != 0) {
// enquanto o número for diferente de 0 adiciona-o à LinkedList

            numbers.add(number);
            number = kbd.nextInt();
        }
        kbd.close();
        return numbers;
    }

/**
 * Print all the elements in the list
 * @param numbers
 */
    public static void printList(List<Integer> numbers) {
        for (Integer integer : numbers) {
            System.out.print(integer + " ");
        }
    }

    public static void usingListMethods() {
        LinkedList<Integer> numbers = addToLinkedList();    // Output Esperado
        printList(numbers);    // 1 2 3 4 5 6 7 8 9
        System.out.println("\nFIRST ELEMENT: " + numbers.getFirst());    //FIRST ELEMENT: 1
        System.out.println("LAST ELEMENT: " + numbers.getLast());    //LAST ELEMENT: 9
        System.out.println("CONTAINS 6: " + numbers.contains(6));    //CONTAINS 6: true
        System.out.println("REMOVE 4: " + numbers.remove(4));    // REMOVE 4: 5
        remover o 4ºelemento da Lista

        numbers.addFirst(234);
        printList(numbers);    // 234 1 2 3 4 6 7 8 9
    }

    public static void main (String[] args){
        usingListMethods();
    }
}
```

Consola:

```
1
2
3
4
5
6
7
8
9
```

```
0
1 2 3 4 5 6 7 8 9
FIRST ELEMENT: 1
LAST ELEMENT: 9
CONTAINS 6: true
REMOVE 4: 5
234 1 2 3 4 6 7 8 9
```

2.

```
package lab19;

import java.util.ArrayList;
import java.util.List;

public class WrongMethod_lab19 {

    public static void wrongMethod(){
        List<Integer> list = new ArrayList<Integer>(); // instanciação de um arrayList
        list.add(1); // adiciona-se vários elementos ao arrayList
        list.add(2);
        list.add(new Integer(4));
        list.add(3);
        list.add(2);
        int sum = 0, count = 0, size = list.size(); //size = 5 (5 elementos)
        for(Integer i : list)
        {
            sum+= i; // somar todos os elementos da lista (1+2+4+3+2= 12 = sum)

            ++count; // count = nº de elementos
        }
        System.out.println("Sum: " + sum);
        System.out.println("Count: " + count);
        System.out.println("Size: " + size);
    }

    public static void main (String[] args){
        wrongMethod();
    }
}
```

Output:

```
Sum: 12
Count: 5
Size: 5
```

3.

```
package lab19;

import java.util.ArrayList;
import java.util.Collection;
import java.util.LinkedList;
import java.util.List;

public class Repetition {
    /**
     * This method will receive two collections of integers and returns a new
     collection whose
     * elements are repeated elements in both collections received
     * @param coll
     * @param col2
     * @return collection whose elements are repeated elements in both collections
     received
     */
}
```

```
public static Collection<Integer> returnRepetitions(Collection<Integer> coll,
Collection<Integer> col2)
{
    Collection<Integer> repetitions = new LinkedList<>();

    // verificar qual das coleções é mais pequena, para otimizar o processo e o tempo de
    // computação
    if (coll.size()<=col2.size())
    {
        Object[] collArray = coll.toArray();
        // transformar a coleção num array para que possam ser comparados todos os elementos de
        // coll com os de col2

        for (int idx =0; idx < collArray.length; idx++)
        {
            if (col2.contains(collArray[idx]))
                repetitions.add((Integer) (collArray[idx]));
        }
        // se a col2 tiver o elemento em análise o mesmo é adicionado à LinkedList, mas tem de
        // ser feito um cast para Integer
    }
    else
    {
        Object[] col2Array = col2.toArray();

        for (int idx =0; idx < col2Array.length; idx++)
        {
            if (coll.contains(col2Array[idx]))
                repetitions.add((Integer) (col2Array[idx]));
        }
    }
    return repetitions;
}
```

```
public static void main(String[] args) {
    List<Integer> coll = new ArrayList<>();
    List<Integer> col2 = new ArrayList<>();
    coll.add(1);
    coll.add(2);
    coll.add(3);
    coll.add(4);
    coll.add(5);
    coll.add(6);
    coll.add(7); // as duas coleções têm em comum os elementos 1, 2, 3, 4
    coll.add(8);
    coll.add(9);
    col2.add(1);
    col2.add(2);
    col2.add(3);
    col2.add(4);
    System.out.println("Repetitions");
    System.out.println(returnRepetitions(coll,col2));
}
```

Output:

Repetitions  
[1, 2, 3, 4]