

Lab 5. Linked Lists

Goal: To learn how to build linked lists.

Part 1 - Designing Singly Linked List

Open the lab project. We will be working with three files: **ListNode**, **LinkedList**, and, of course, **TestSuite**.

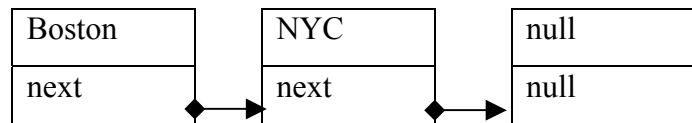
A linked list is defined as a list of nodes, where each node contains some data, and a link (reference) to the next node of the list. Read the file **ListNode**. The only interesting part in the file is the definition of the sentinel node.

The definition is:

```
public static ListNode sentinel = new ListNode(null, null);
```

Defining a member data as **static** means that the value is set once and for all and can never be changed. we call this type of data 'constants'. This particular constant will be used to mark the end of the list.

The class **LinkedList** contains the definition of the linked list. Its only member data is a link to the first item in the list. Empty list contains one node - the sentinel node. New node is added at the front, connecting its next field to the node which was the first in the original list.



Make sure you understand how the list is built.

Part 2. Using the Linked List

- Define a **LinkedList** iterator, which implements the **IRange** interface for this linked list.
- Design **findCity** method for this list, using the **LinkedList** iterator, looking up the city by the city name.
- Design the method **countZipcodes**, which counts the number of different zip codes for the city with the given name.
- Design the method **sortedInsert**, which adds the given city into this sorted list, in the appropriate place. (Again, the sorting is done by the name of the city - do not worry about duplicate cities - or the zip codes out of order.)
- Optionally, design a sorting algorithm, which will create a sorted list by removing one item at a time from an old list and inserting it in sorted order into a new list - at the end changing the reference to first to the first item in the sorted list.