

Lab 7: Singly-Linked Lists

Objective

This lab will explore the implementation of a singly linked list in Java. Linked lists are an implementation tool they are used to implement other collection objects. In this lab, we implement the List interface using a singly linked list. The singly linked list class we create will be housed in a cscollections sub-package of the cs package in our CSLibrary. A secondary objective of this lab is to have you review the methodology we used to implement and test a collection in Lab 3. You should follow the same methodology in this lab.

Create a Singly-Linked List Collection

The design of the singly-linked list collection is given in Figure 1 below. Note package structure. The SinglyLinkedList class is in the cscollections package, which is nested in the cs package.

Figure 1 also indicates that Node is self-referential in the sense that it has an instance variable of type Node. This is indicated by the association line that loops from the Node class to the Node class on the upper right corner. It graphically represents the relationship created by the declaration `-next: Node<E>`.

There is no new repo for this lab. Work in your CSLibrary repo from Lab 4. Push the code for this lab back to the master branch of that repo. (Do not push this code to the Lab-4 branch of the repo.) Implement the package hierarchy by adding the cscollections package to your cs package. Then implement the List interface, the SinglyLinkedList class, and the Node class.

Implement the List interface first. Add the Node class with stubs for each method. Implement the Node constructors and methods one at a time. After the Implement the SinglyLinkedList class with stubs for each method. Implement the following methods in the SinglyLinkedList class:

- the constructor
- isEmpty
- getLength
- prepend
- append

Lab 9: Singly-Linked Lists

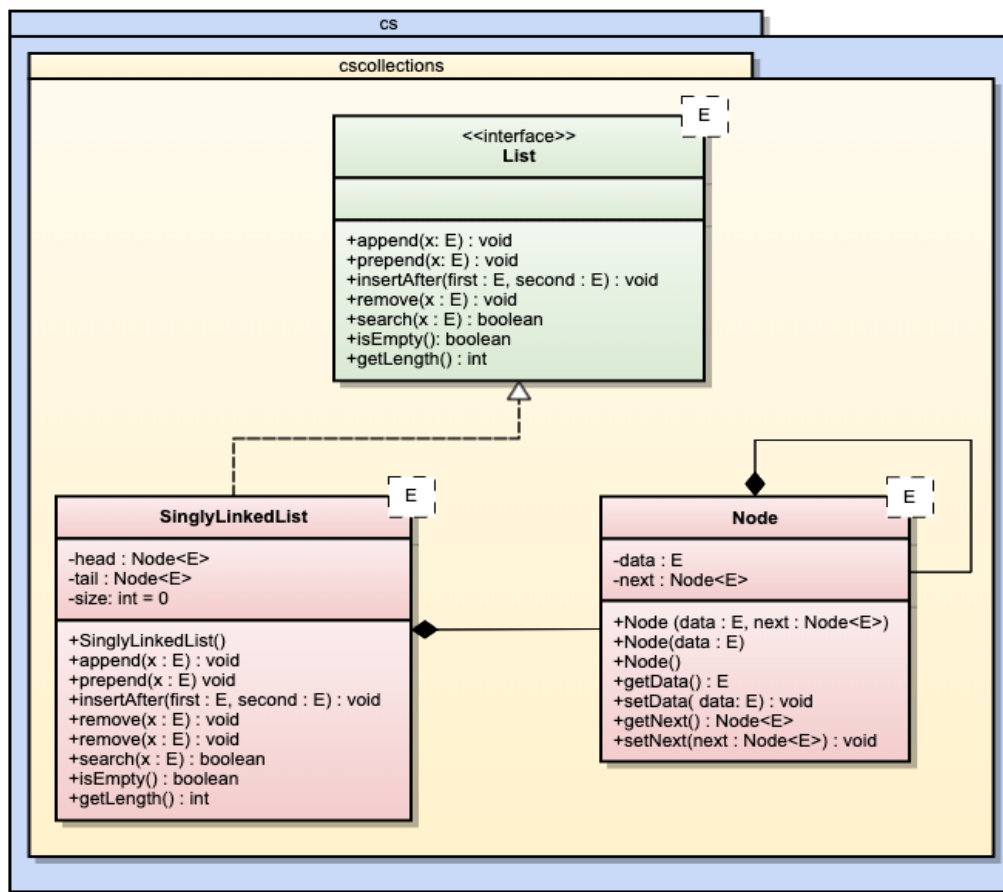


Figure 1: Design of a SinglyLinkedList ADT

Start testing by coding the constructor for SinglyLinkedList. Inspect your code to ensure it is correct. Do the same for isEmpty and getLength. Finally implement prepend and append in that order. Test each as you implement them following the procedure illustrated in the lab lecture. Do not forget to increment the size whenever a node is added.¹

Finishing up

When you have your Singly-Link List collection created and tested, commit your completed lab code to your local repository and push the commit back to your GitHub repository.

¹ Do not implement insertAfter and search. We will implement these methods in the next lab.
Department of Math and Computer Science
V2.2 October 5, 2020
S. Sigman