COSC 1437 Programming Lab & Assignment (Fall 2016)

List referenced-based & Extra

Topics

- The main topic in Ch. 5 is a reference-based implementation of singly linked list. Reference-based singly linked list is the second abstract data type (ADT) and it will be reused in later chapters, including Ch.7 (Stack), Ch.8 (Queues), to implement advanced ADT. Therefore, it is important to clearly understand the implementation details.
- Through this lab, you are asked to complete the reference-based list, plus several other methods that are frequently used in real applications. Furthermore, you will practice implementing recursive methods.
- In addition, you are asked to test your implementation with a driver, called "TestListReferenceBased.java", in which main() calls the implemented methods and validate your implementation.

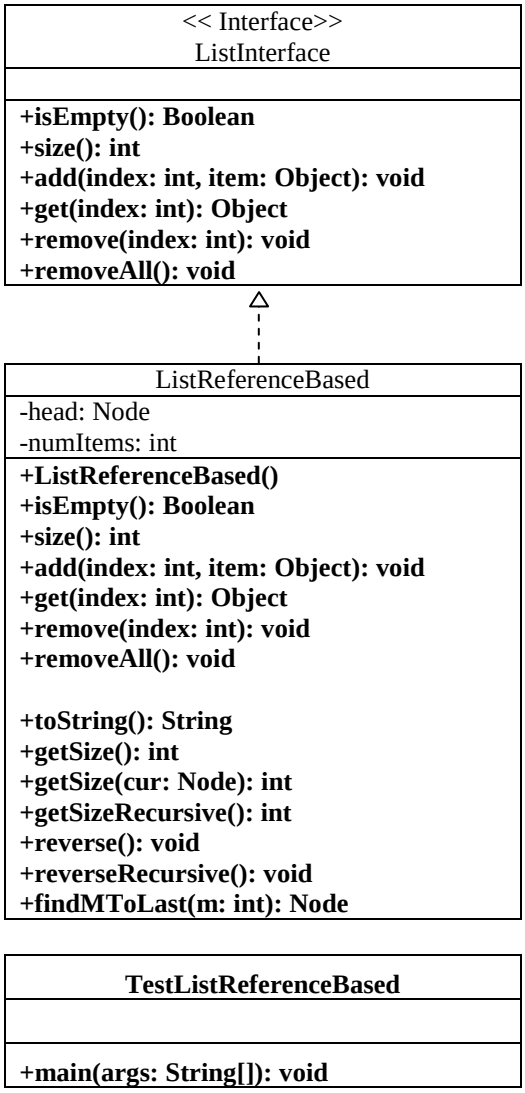
Main Activitie

- First, you are asked to implement "Node" class as shown in UML below.

Node
-item: Object -next: Node
+Node(newItem: Object) +Node(newItem: Object, nextNode: Node) +setItem(newItem: Object): void +getItem(): Object +setNext(nextNode: Node): void +getNext(): Node

- Then, you are asked to complete "ListReferenceBased" class, which implements "ListInterface" interface as shown in the UML in the next page. Note that "ListInterface.java" is given and you can just use as it is.
- Many methods have been already implemented in ListReferenceBased.java; thus, please make sure you fully understand details of each method, referring to the lecture slide for the implementation details.
- The remaining methods are more challenging, but it will give you a great opportunity to review your understanding on recursion and also to practice how to efficiently manipulate references.
- You are given the following files:
  - ListException.java
  - ListIndexOutOfBoundsException.java
  - ListInterface.java
  - **ListReferenceBased.java**
  - **Node.java**
  - TestListReferenceBased.java

  and the lab description pdf file:
  - ListReferenceBased.pdf (➔ this file)



- NOTE: We assume that every index always starts from 0, not 1.
- Description of each method is given in the code. Also, detail agenda will be discussed in the lab.

Optional Activities (for Extra Points)

- N/A

What to Hand in

- Turn in your programs (i.e., "**ListReferenceBased.java**") via Blackboard.