

Due Sunday February 14th, 2016 at 11:55PM

This assignment is the first portion of the List project for this course. The primary goals are to compare the ArrayList and LinkedList implementations of the List ADT and to become familiar with generic type arguments and list operations.

Overview

The primary goals of this program are as follows:

- (1) Complete the implementation of the LinkedList and ListNode classes.
- (2) Compare the LinkedList and ArrayList implementations using the Stopwatch class from Program 0.1

Download the List interface, the AList implementation, and the skeleton code for the ListNode and LinkedList classes from Moodle. Copy or import the Timer interface and Stopwatch class from Program 0.1. Create a new Driver class.

AList is completely implemented based on the code from the textbook.

Timer and **StopWatch** should be unchanged from Program 0.1.

LinkedList and **ListNode** contain method signatures and commented instructions. We will begin implementing these methods during lab on February 11th, and you should finish the implementation with code from the textbook. *See Figures 4.4 and 4.8 for details.*

Driver should create instances of the Stopwatch, AList, and LinkedList classes. You will compare execution times for the following three operations: append, insert, and moveToPos.

Tasks

- Your Driver should create instances of the Stopwatch, AList, and LinkedList classes.
- Both List classes should contain the same Type argument. Integer is the most straightforward choice.
- Time how long it takes to append 100,000 items to each list. Items can have any value you like. Randomly generating values is an easy option.
- Clear both lists.
- Time how long it takes to insert 100,000 items to each list.
- Time how long it takes to move to a random position 100,000 times in each list.
- Report the average result for each operation after running it at least 3 times.

	append	insert	moveToPos
ArrayList			
LinkedList			

Scoring Rubric

Driver:

25 points

- Up to 5 points for appropriate documentation and comments
- Up to 5 points for correctly appending 100,000 items to both lists
- Up to 5 points for correctly inserting 100,000 items in both lists
- Up to 5 points for correctly moving to random positions 100,000 times in both lists
- Up to 5 points for adequately reporting timing results for each operation

Lists:

15 points

- Up to 10 points for correctly implementing the Linked List class
- Up to 5 points for correctly implementing the Linked Node class
-

Total points available: 40

This will be graded as a programming assignment

Late programs turned in prior to class on Tuesday (February 16th) will receive a 5 point penalty.

Programs turned in after class on Tuesday (February 16th) will receive at most 60% of the available points.