CSCI 2235
Programming Assignment 02 - Lists

Assigned: September 23, 2019
Due: October 04, 2019 @ 23:00h

Purpose

- To explore the construction of List Structures
- To explore the use of List Structures in an application

Problem Statement

We need to design a scoreboard to store highscores for a video game. You are given the classes which represent a score, called GameEntry, and the Scoreboard. Each game entry simply contains the score associated with a player's name. The scoreboard maintains a list of scores which has a maximum capacity that can be saved. It is your job to implement the underlying data structure which will be used to store each score. In this case we will be implementing list structures to store the scores. Each list will need to implement, and conform to, a provided interface called List. The capacity constraint will be a part of the Scoreboard class and will be handled within that classes methods.

To get you started a gradle project has been created. Within the source directly you will find two main packages: edu.isu.cs2235.structures and edu.isu.cs2235.structures.impl. The former package is the location of the interface List and the latter is the location for the implementation SinglyLinkedList and DoublyLinkedList. Outside of these packages are three other files Driver.java which contains the executable main method for the program; GameEntry.java which contains the GameEntry class; and Scoreboard.java which contains the Scoreboard class. You will need to fill in the method bodies for the add and remove methods of the Scoreboard class. I leave the complete implementation of the SinglyLinkedList and DoublyLinkedList up to you.

Assignment

1. Build a generic SinglyLinkedList class which implements the List<E> interface, found in package edu.isu.cs2235.structures in a class called SinglyLinkedList<E> in the new package edu.isu.cs2235.structures.impl. This class must have the following methods:

   - **public** E first ()
   - **public** E last ()
   - **public void** addLast(E element)
   - **public void** addFirst(E element)

- **public** E removeFirst()
- **public** E removeLast()
- **public void** insert(E element, **int** index)
- **public** E remove(**int** index)
- **public** E get(**int** index)
- **public int** size()
- **public boolean** isEmpty()
- **public void** printList()

2. Build a generic DoublyLinkedList class which implements the List<E> interface, found in package edu.isu.cs2235.structures, in a class called DoublyLinkedList<E> in the new package edu.isu.cs2235.structures.impl. This class must have the following methods:

   - **public** E first ()
   - **public** E last ()
   - **public void** addLast(E element)
   - **public void** addFirst(E element)
   - **public** E removeFirst()
   - **public** E removeLast()
   - **public void** insert(E element, **int** index)
   - **public** E remove(**int** index)
   - **public** E get(**int** index)
   - **public int** size()
   - **public boolean** isEmpty()
   - **public void** printList()

3. In the Scoreboard class there are three methods add(GameEntry entry) and remove(**int** index) which are required to be implemented. Look at the associated documentation comments to understand the requirements. Note that the add method must meet the following requirements:

   - Adds items up to a maximum capacity (as stored in the Scoreboard capacity field).
   - Items are added in descending order based on the score field of the GameEntry class.

4. Associated with each class are a set of unit tests. You should execute the unit tests for each component in order to ensure that you have correctly completed the assignment.

   To understand the requirements for each method look at the documentation in the edu.isu.cs2235.structures.List **interface**. You must ensure that each of the requirements for each method are correctly implemented.

   Your implementation of Scoreboard will need to use one of the two lists, I leave the selection of the type up to you.

Submission

When you have completed the assignment (all tests pass) or it is reaching midnight of the due date, make sure you have completed the following:

0. Your code compiles with no compilation errors
1. Zip your entire project directory into a file named [firstname]_[lastname].zip (Note that the square brackets are not part of the file name but simply indicate required information)
2. Submit the zip file to moodle in the PA02 dropbox.

Grading – 50 Points

- Singly Linked List:

    - **public** E first () implemented correctly – 1 points
    - **public** E last () implemented correctly – 1 points
    - **public void** addLast(E element) implemented correctly – 2 points
    - **public void** addFirst(E element) implemented correctly – 1.5 points
    - **public** E removeFirst() implemented correctly – 1 points
    - **public** E removeLast() implemented correctly – .5 points
    - **public void** insert(E element, **int** index) implemented correctly – 2.5 points
    - **public** E remove(**int** index) implemented correctly – 2 points
    - **public** E get(**int** index) implemented correctly – 3 points
    - **public void** printList() implemented correctly – 1 points

- Doubly Linked List:

    - **public** E first () implemented correctly – 1 points
    - **public** E last () implemented correctly – 1 points
    - **public void** addLast(E element) implemented correctly – 2 points
    - **public void** addFirst(E element) implemented correctly – 1.5 points
    - **public** E removeFirst() implemented correctly – 1 points
    - **public** E removeLast() implemented correctly – .5 points
    - **public void** insert(E element, **int** index) implemented correctly – 2.5 points
    - **public** E remove(**int** index) implemented correctly – 2 points
    - **public** E get(**int** index) implemented correctly – 3 points
    - **public void** printList() implemented correctly – 1 points

- Scoreboard class

    - add method implemented correctly – 6 points
    - remove method implemented correctly – 8 points

- Followed good programming practice:

    - Program by Contract - 2 points
    - Code is Commented 3 points

Hints

- The basic implementation of the Singly and Doubly linked lists are defined in the book in Chapter 3. I would consult this prior to beginning implementation.
- The basic implementation (using arrays) for the Scoreboard add() and remove() methods are depicted in the code fragments 3.3 and 3.4 in the book. Use only as a reference but not as an implementation guideline.