Design Diagram Assignment 3

John Mobarry | March 7, 2021

Class Node

data: T-previous: Node-next: Node

+Node(T data, Node previous, Node next):

Paramaterized constructor with data, previous, and next +Node(T data):

Constructor for an unlinked node +setData(T data) : void set the data of a node to the specified data

+getData(): T

return the data of the current node +setNext(Node n): void the the next node to the entered node

+getNext(): Node

return the node that is the next in the chain

+hasNext(): boolean

return wether the current node has a node in the next chain

+setPrevious(Node n):

set the previous node to the entered node

in the method

+getPrevious(): node

return the previous node in the chain +hasPrevious(): Boolean

return wether the current node has a node

in the previous in the chain

Class Mylterator implements ListIterator<T>

+current: Node +atEnd: boolean +atStart: boolean

+MyIterator():

default constructor of the iterator that sets

current = null

+hasNext(): boolean

+hasPrevious(): boolean

+next():

traverses to the next in the iterator +previous():

traverses to the previous in the iterator

Class BasicDoubleLinkedList<T>

-size: int-FirstNode: Node-LastNode: Node

+ BasicDoubleLinkedList():

No arg constructor to create an empty list +addToEnd(T data):

Adds an element to the end of the list

+addToFront(T data):

Adds an element to the front of the list +getFirst(): T

Returns but does not remove the efirst element from the list +qetLast(): T

Returns but does not remove the last element from the list +getSize(): int

traverse the list, compute the size, and return the size

+iterator():mylterator method that is implemented using an inner class that implements ListIterator and defines the methods of hasNext(), next(), hasPrevious(), and previous()

+remove(T targetData, java.util.Comparator<T> comparator): void

Removes the first instance of the targetData from the list

targetData from the list +retrieveFirstElement():T

Removes and returns the first element from the list

+retreieveLastElement(): T

Removes and returns the last element from the list

+toArrayList(): ArrayList<T>:

Returns an arraylist of the items in the list from head of list to the tail of the list

Class SortedDoubleLinkedList<T> extends
BasicDoubleLinkedList<T>

+ comparator: Comparator

+SortedDoubleLinkedList(Comparator c): constructor that creates an instance of the class using the super() and setting comparator = c;

+add(): void

Inserts the specified element at the correct position in the sorted list +addToEnd(T data): void @override and not implemented +addToFront(T data): void @override and not implemented

+iterator(): implements the iterator by calling the superclass iterator method +remove(T data, Comparator comparator):

Implements the remove uperation by calling the super class remove method