Module 6 Lab 6A

Name: Clayton Black
Date: 11-14-2019
Assignment Name: Module 6 Lab 6B
Assignment Brief: Linked List Sources:

- https://ccse.kennesaw.edu/fye/pseudocode/pseudocodeguide.php

Main

```
CLASS Main
BEGIN

    METHOD Main
    BEGIN
        CREATE t1 ← NEW Trivia("Trivia Game 1", 1, 1.0, 2)
        CREATE t1 ← NEW Trivia("Trivia Game 2", 2, 1.0, 2)
        CREATE t1 ← NEW Trivia("Trivia Game 3", 3, 1.0, 2)
        CREATE t1 ← NEW Trivia("Trivia Game 4", 4, 1.0, 2)
        CREATE t1 ← NEW Trivia("Trivia Game 5", 5, 1.0, 2)

        CREATE ll ← NEW TriviaLinkedList()
        PRINT ll
        printLinkedList(ll)

        addAndPrint(ll, t5)
        addAndPrint(ll, t4)
        addAndPrint(ll, t3)
        addAndPrint(ll, t2)
        addAndPrint(ll, t1)

        PRINT "Remove node by triviaGameID"
        ll.removeByID(3);
        printLinkedList(ll)

        PRINT "Remove node that doesn't exist"
        removed ← ll.removeByID(23)
        IF (removed == null) THEN
            PRINT "Node Does Not Exist"
        ENDIF
        printLinkedList(ll)
    END METHOD

    METHOD printLinkedList
    BEGIN
        current = ll.head
        PRINT "\n{\n"
        WHILE (current != null){
            print "\t" + current.element
            current = current.next
        ENDWHILE

        print "}\n\n"

    END METHOD

    METHOD addAndPrint(ll, t)
    BEGIN
        ll.insert(t)
        PRINT ll
        printLinkedList(ll)
    END METHOD
END CLASS
```

Game

```
CLASS Game
BEGIN
```

```
    CREATE description;

    CONSTRUCTOR
    BEGIN
        THIS("This Game has no description")
    END CONSTRUCTOR

    CONSTRUCTOR(parameters: description)
    BEGIN
        this.description ← description
    END CONSTRUCTOR

    METHOD getDescription()
    BEGIN
        RETURN description
    END METHOD

    METHOD setDescription(parameters: description)
    BEGIN
        this.description ← description
    END METHOD

    METHOD toString()
    BEGIN
        RETURN "Game{" +
                "description='" + description + '\'' +
                '}'
    END METHOD

END CLASS
```

Trivia

```
CLASS Trivia EXTENDS Game
    CREATE description, triviaGameID, ultimatePrizeMoney, numberOfQuestionsThatMustBeAnsweredToWin

    CONSTRUCTOR (parameters: description, triviaGameID, ultimatePrizeMoney, numberOfQuestionsThatMustBeAnsweredToWin)
    BEGIN
        super(description)
        this.triviaGameID ← triviaGameID
        this.ultimatePrizeMoney ← ultimatePrizeMoney
        this.numberOfQuestionsThatMustBeAnsweredToWin ← numberOfQuestionsThatMustBeAnsweredToWin
    END CONSTRUCTOR

    METHOD getTriviaGameID()
    BEGIN
        RETURN triviaGameID;
    END METHOD

    METHOD setTriviaGameID(parameters: triviaGameID)
    BEGIN
        this.triviaGameID ← triviaGameID
    END METHOD

    METHOD getUltimatePrizeMoney()
    BEGIN
        RETURN ultimatePrizeMoney
    END METHOD

    METHOD setUltimatePrizeMoney(parameters: ultimatePrizeMoney)
    BEGIN
        this.ultimatePrizeMoney ← ultimatePrizeMoney
    END METHOD

    METHOD getNumberOfQuestionsThatMustBeAnsweredToWin()
    BEGIN
        RETURN numberOfQuestionsThatMustBeAnsweredToWin
    END METHOD

    METHOD setNumberOfQuestionsThatMustBeAnsweredToWin(int numberOfQuestionsThatMustBeAnsweredToWin)
    BEGIN
        this.numberOfQuestionsThatMustBeAnsweredToWin ← numberOfQuestionsThatMustBeAnsweredToWin
    END METHOD

    METHOD toString()
    BEGIN
```

```
        RETURN "Trivia{" +
                "triviaGameID=" + triviaGameID +
                ", ultimatePrizeMoney=" + ultimatePrizeMoney +
                ", numberOfQuestionsThatMustBeAnsweredToWin=" + numberOfQuestionsThatMustBeAnsweredToWin +
                ", description='" + description + '\'' +
                '}'
    END METHOD
END CLASS
```

TriviaNode

```
CLASS TriviaNode
BEGIN
    CREATE element, next

    CONSTRUCTOR (element)
    BEGIN
        this.element ← element
    END CONSTRUCTOR

    METHOD getElement()
    BEGIN
        RETURN element
    END METHOD

    METHOD getNext()
    BEGIN
        RETURN next
    END METHOD

    METHOD setNext()
    BEGIN
        this.next = next
    END METHOD

    METHOD toString()
    BEGIN
        return "TriviaNode{" +
                "element=" + element +
                ", next=" + next +
                '}'
    END METHOD
END CLASS
```

TriviaLinkedList

```
public class TriviaLinkedList {
    TriviaNode head, tail;
    private int size = 0;

    public TriviaLinkedList() {
    }

    public void addFirst(Trivia e){
        TriviaNode newNode = new TriviaNode(e);
        newNode.next = head;
        head = newNode;
        size++;

        if (tail == null){
            tail = head;
        }
    }

    public void addLast(Trivia e){
        TriviaNode newNode = new TriviaNode(e);

        if (tail == null){
            head = newNode;
            tail = newNode;
        } else {
            tail.next = newNode;
            tail = newNode;
```

```
        }

        size++;
    }

    public void insert(Trivia e){
        insert(0, e);
    }

    public void insert(int index, Trivia e){
        if (index == 0){
            addFirst(e);
        } else if (index >= size){
            addLast(e);
        } else {
            TriviaNode cur = head;
            for (int i = 1; i < index; i++) {
                cur = cur.next;
            }
            TriviaNode tmp = cur.next;
            cur.next = new TriviaNode(e);
            (cur.next).next = tmp;
            size++;
        }
    }

    public Trivia removeFirst(){
        if (size == 0)
            return null;
        else {
            TriviaNode temp = head;
            head = head.next;
            size--;
            if(head == null){
                tail = null;
            }
            return temp.element;
        }
    }

    public Trivia removeLast(){
        if (size == 0){
            return null;
        } else if (size == 1){
            TriviaNode tmp = head;
            head = null;
            tail = null;
            size = 0;
            return tmp.element;
        } else {
            TriviaNode cur = head;

            for (int i = 0; i < size - 2; i++) {
                cur = cur.next;
            }

            TriviaNode tmp = tail;
            tail = cur;
            tail.next = null;
            size--;
            return tmp.element;

        }
    }

    public Trivia remove(int index){
        if (index < 0 || index >= size){
            return null;
        } else if (index == 0){
            return removeFirst();
        } else if (index == size - 1){
            return removeLast();
        } else {
            TriviaNode pre = head;

            for (int i = 1; i < index; i++) {
                pre = pre.next;
            }
```

```
            TriviaNode cur = pre.next;
            pre.next = cur.next;
            size--;

            return cur.element;
        }
    }

    public Trivia removeByID(int id){
        TriviaNode cur = head;
        int index = 0;
        while(cur != null){
            if (id == cur.element.getTriviaGameID())
                return remove(index);
            else
                cur = cur.next;
            index++;
        }

        return null;
    }

    @Override
    public String toString() {
        return "TriviaLinkedList{" +
                "head=" + head +
                ", tail=" + tail +
                ", size=" + size +
                '}';
    }
}
```