

ΕΡΓΑΣΤΗΡΙΟ 5

Υλοποίηση Stack και Queue με χρήση της SimpleLinkedList

Στην άσκηση αυτή θα χρησιμοποιήσετε τις κλάσεις του κόμβου και της απλά συνδεδεμένης λίστας που υλοποιήσατε στο προηγούμενο εργαστήριο (άσκηση 4).

- **ListNode class** η οποία περιγράφει ένα κόμβο μιας απλά συνδεδεμένης λίστας
- **SimpleLinkedList class** που υλοποιεί το **List interface**

Άσκηση 5.1

Να δώσετε μια υλοποίηση του **Stack interface** με χρήση της **SimpleLinkedList**

Στη συνέχεια να ελέγξετε/ τροποποιήσετε το πρόγραμμα της άσκησης «για επιπλέον εξάσκηση» ώστε να δουλεύει με την καινούργια υλοποίηση της στοίβας. (Δίνεται εκ νέου στη συνέχεια)

Να γραφεί πρόγραμμα Java για τον έλεγχο της σωστής χρήσης των παρενθέσεων () σε μία αριθμητική παράσταση χρησιμοποιώντας τη δομή δεδομένων στοίβα, η οποία θα υλοποιείται με **απλά συνδεδεμένη λίστα**. Το πρόγραμμα να επιστρέφει μήνυμα εάν η αριθμητική παράσταση είναι σωστή ή εάν έχει λάθος, να αναφέρει το σημείο που βρήκε το λάθος.

Άσκηση 5.2

Να δώσετε μια υλοποίηση του **Queue interface** με χρήση της **SimpleLinkedList**

Στη συνέχεια να ελέγξετε/ τροποποιήσετε το πρόγραμμα της άσκησης «για επιπλέον εξάσκηση» ώστε να δουλεύει με την καινούργια υλοποίηση της ουράς. (Δίνεται εκ νέου στη συνέχεια)

Να γραφεί πρόγραμμα Java για την εξυπηρέτηση αυτοκινήτων σε διόδια με την χρήση **ουράς**, η οποία θα υλοποιείται με **απλά συνδεδεμένη λίστα**. Πιο συγκεκριμένα θα εμφανίζεται το παρακάτω μενού:

MENΟΥ

1. Άφιξη αυτοκινήτου
2. Αναχώρηση αυτοκινήτου
3. Κατάσταση ουράς
4. Έξοδος

Επιλογή 1: Θα πληκτρολογούνται τα στοιχεία του αυτοκινήτου π.χ. ο αριθμός αυτοκινήτου και θα τοποθετείται στο τέλος της ουράς.

Επιλογή 2: Το αυτοκίνητο που βρίσκεται πρώτο στην ουρά θα διαγράφεται μαζί με ένα ανάλογο μήνυμα επιβεβαίωσης.

Επιλογή 3: Θα εμφανίζονται με τη σειρά οι αριθμοί των αυτοκινήτων που παραμένουν στην ουρά για να εξυπηρετηθούν.

Επιλογή 4: Το πρόγραμμα θα τερματίζεται.

Δίνονται επίσης:

Αρχείο Stack.java

```
public interface Stack
{
    public int size();
    public boolean isEmpty();
    public boolean isFull();
    public Object top() throws StackEmptyException;
    public void push(Object item) throws StackFullException;
    public Object pop() throws StackEmptyException;}
```

Αρχείο Queue.java

```
public interface Queue
{
    public int size();
    public boolean isEmpty();
    public boolean isFull();
    public Object front() throws QueueEmptyException;
    public void add(Object item) throws QueueFullException;
    public Object remove() throws QueueEmptyException; }
```

Αρχείο ListNode.java

```
public class SLListNode
{
    // Instance fields
    private Object data;
    private SLListNode nextNode;

    // Instance methods
    public Object getNodeData()
    public SLListNode getNextNode()
    public void setNodeData(Object d)
    public void setNextNode(SLListNode n)
    public String toString()
}
```

Αρχείο List.java

```
public interface List
{
    public boolean isEmpty();
    public int size();
    public void insertFirst(Object data);
    public void insertLast(Object data);
    public Object removeFirst() throws ListEmptyException;
    public Object removeLast() throws ListEmptyException; }
```

Αρχείο ListEmptyException.java

```
public class ListEmptyException extends RuntimeException
{
    public ListEmptyException(String err)
    { super(err); }
}
```