

ΕΡΓΑΣΤΗΡΙΟ 4

Δημιουργία και χρήση απλά συνδεδεμένης λίστας (*SimpleLinkedList*)

Άσκηση 4.1

Για να κατασκευάσετε μια απλά συνδεδεμένη λίστα πρέπει να δημιουργήσετε τις δύο παρακάτω κλάσεις:

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

Δίνεται - ημιτελής - η κλάση ***SLListNode*** και το **List** interface:

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); }
}
```

Άσκηση 4.2

Να γραφεί πρόγραμμα (***SLListManagement.java***) για την διαχείριση μίας λίστας, η οποία περιέχει Strings. Η διαχείριση να γίνεται μέσω ενός μενού όπως το παρακάτω:

******* LINKED LIST MANAGEMENT *******

- 1- INSERT ELEMENT AT THE BEGINNING OF THE LIST
- 2- INSERT ELEMENT AT THE END OF THE LIST
- 3- DELETE FROM THE BEGINNING OF THE LIST
- 4- DELETE FROM THE END OF THE LIST
- 5- LIST LENGTH
- 6- IS THE LIST EMPTY
- 7- PRINT LIST
- 8- EXIT

INPUT YOUR CHOICE (π.χ. 5):

Επιπλέον Εξάσκηση:

Δημιουργήστε μία απλά συνδεδεμένη λίστα η οποία να περιέχει σπουδαστές με τον βαθμό που έχουν πετύχει στις εξετάσεις του μαθήματος (μπορείτε να χρησιμοποιήσετε την κλάση **Student** από προηγούμενο εργαστήριο).

Στη συνέχεια γράψτε πρόγραμμα το οποίο να βρίσκει και να επιστρέφει (**α**) το σπουδαστή που πέτυχε τη μεγαλύτερη βαθμολογία και (**β**) το σπουδαστή με τη μικρότερη βαθμολογία. (Μπορεί μία μέθοδος να επιστρέφει και τους δύο σπουδαστές ταυτόχρονα;)