

Δομές Δεδομένων & Ανάλυση Αλγορίθμων 3ο Εξάμηνο

Διπλά Συνδεδεμένες Λίστες (Double Linked Lists)

Δημοσθένης Σταμάτης

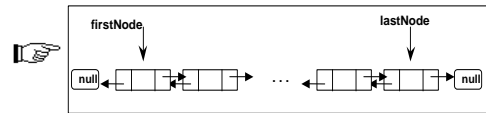
<http://www.ise.ihu.gr/~demos>

Τμήμα Μηχανικών Πληροφορικής & Ηλεκτρονικών Συστημάτων

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

Αναπαράσταση Διπλά Συνδεδεμένης Λίστας



Διπλά Συνδεδεμένη λίστα με πρόσβαση στην αρχή και στο τέλος

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ



```
class DoubleNode
//Ορισμός κόμβου μιας διπλά συνδεδεμένης λίστας (double linked list)
{
    private Object item;
    private DoubleNode next, previous;

    public DoubleNode( ) {
        this(null,null,null);
    }

    public DoubleNode(Object it, DoubleNode n, DoubleNode p) {
        item = it;
        next = n;
        previous = p;
    }

    public void setItem(Object newItem) {
        item = newItem;
    }
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

```
class DoubleNode
```

```
    public void setNext(DoubleNode newNext) {
        next = newNext;
    }
    public void setPrevious(DoubleNode newPrevious) {
        previous = newPrevious;
    }
    public Object getItem( ) {
        return(item);
    }
    public DoubleNode getNext( ) {
        return(next);
    }
    public DoubleNode getPrevious( ) {
        return(previous);
    }
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

```
public class DoubleLinkedList
//Υλοποίηση μίας διπλά συνδεδεμένης λίστας (double linked list)
{
    private DoubleNode firstNode, lastNode;
    private int size;
    // Αναπαράσταση της λίστας με τη βοήθεια δύο δεικτών σε 1ο και τελευταίο κόμβο

    public DoubleLinkedList( ) {
        firstNode = lastNode = null;
        size = 0;
    }
    public DoubleNode getFirst( ) {
        return firstNode;
    }
    public DoubleNode getLast( ) {
        return lastNode;
    }
    public boolean isEmpty( ) {
        return(firstNode == null);
    }
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

```
    public int size( ) {
        return size;
    }
    public void insertFirst( Object newItem) {
        if (isEmpty( ))
            firstNode = lastNode = new DoubleNode(newItem, null, null);
        else {
            DoubleNode B = new DoubleNode(newItem, firstNode, null);
            firstNode.setPrevious(B);
            firstNode = B
        }
        size++;
    }
    public void insertLast( Object newItem) {
        if (isEmpty( ))
            firstNode = lastNode = new DoubleNode(newItem, null, null);
        else {
            DoubleNode B = new DoubleNode(newItem, null, lastNode);
            lastNode.setNext(B);
            lastNode = B;
        }
        size++;
    }
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

```
public Object removeFirst( ) throws ListEmptyException
{
    Object removeItem;
    if (isEmpty( ))
        throw new ListEmptyException("Empty List!!!");
    removeItem = firstNode.getItem( );
    if (firstNode == lastNode)
        firstNode = lastNode = null;
    else {
        DoubleNode B = firstNode.getNext( );
        firstNode.setNext(null); //χρηιάζεται?
        B.setPrevious(null);
        firstNode = B;
    }
    size--;
    return removeItem;
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

```
public Object removeLast( ) throws ListEmptyException
{
    Object removeItem;
    if (isEmpty( ))
        throw new ListEmptyException("Empty List!!!");
    removeItem = lastNode.getItem();
    if (firstNode == lastNode)
        firstNode = lastNode = null;
    else {
        DoubleNode B = lastNode.getPrevious( );
        lastNode.setPrevious(null); //?
        B.setNext(null);
        LastNode = B;
    }
    size--;
    return removeItem;
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ