ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ
Εργασία 1

ΜΑΓΚΟΣ ΡΑΦΑΗΛ-ΓΕΩΡΓΙΟΣ 3100098

ΜΠΟΓΔΑΝΟΣ ΜΙΧΑΗΛ 3100123

Α.Επεξήγηση της υλοποίησης των διεπαφών

1)Τάξη IntQueueImpl

-Private αναφορές head και tail που θα δείχνουν στο πρώτο και στο τελευταίο στοιχείο αντίστοιχα.

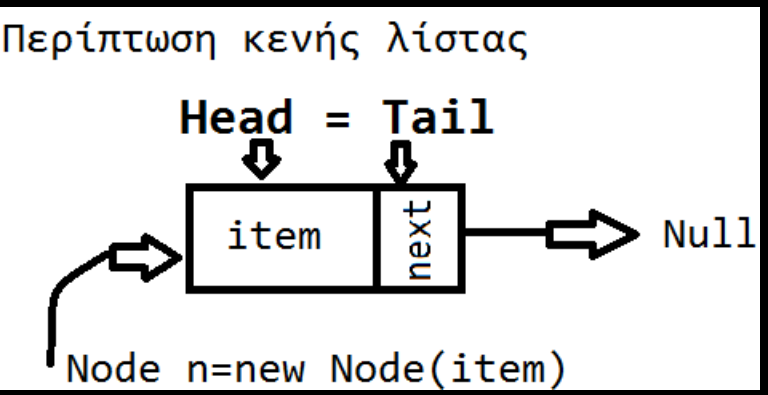
-Private inner class Node που περιλαμβάνει private μεταβλητή int val για την εκχώρηση τιμής στο Node,private αναφορά Node next για να εκχωρούμε τον επόμενο κόμβο στον οποίο θα δείχνει το τρέχον αντικείμενο Node,κατάλληλες μεθόδους get για επιστροφή της τιμής του κόμβου και του επόμενου κόμβου, 2 κατασκευαστές εκ των οποίων ο πρώτος δέχεται 2 ορίσματα,την τιμή του κόμβου και τον επόμενο κόμβο και ο δεύτερος κατασκευαστής 1 όρισμα ,συγκεκριμένα την τιμή του κόμβου , και καλέι τον κατασκευαστή 2 ορισμάτων με παραμέτρους την τιμή του κόμβου και την τιμή null για τον επόμενο κόμβο.

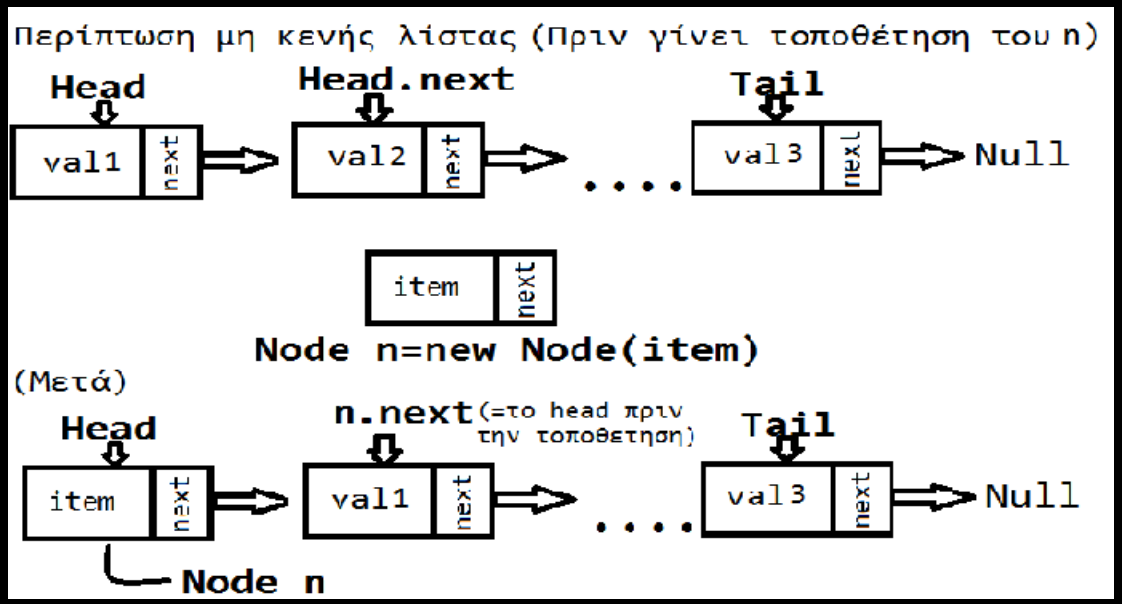
-Μέθοδοι getHead() και getTail() για επιστροφή των private αναφορών head και tail αντίστοιχα.

-Μέθοδος isEmpty() που ελέγχει εάν η λίστα είναι κενή εξετάζοντας την αναφορά head αν είναι null.

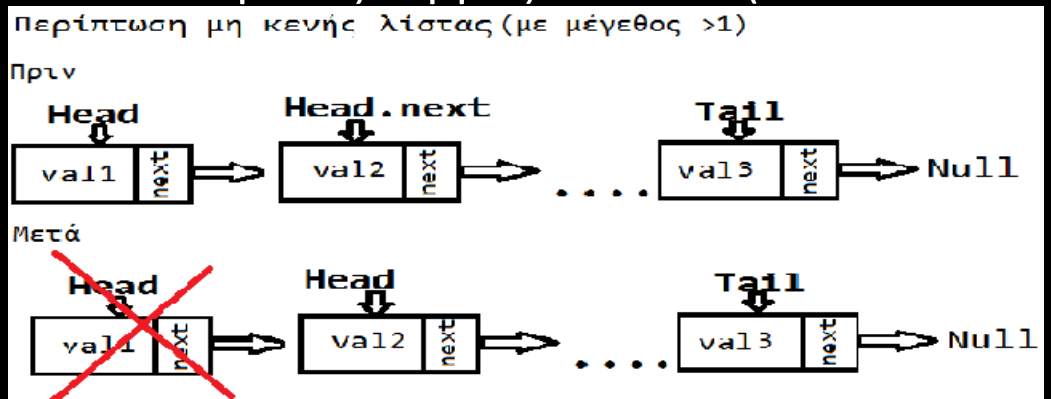
-Μέθοδος put(int item) αρχικά δημιουργεί ένα νέο αντικείμενο Node εκχωρόντας στον κατασκευαστή ενός ορίσματος το int item(άρα θέτει το επόμενο Node που θα δείχνει ο νέος κόμβος ίσο με null).Εάν η λίστα είναι άδεια(κλήση της isEmpty()) τότε και οι δύο αναφορές head και tail γίνονται ίσες με τον νέο κόμβο.Αλλιώς το πεδίο next του νέου κόμβου γίνεται ίσο με head και επίσης το head πλέον δείχνει στον νέο κόμβο,ώστε το στοιχείο να τοποθετηθεί στην αρχή της λίστας.Η μέθοδος ολοκληρώνεται σε O(1) πράξεις κάθε εντολή εκτελείται το πολυ λίγες φορές(και ανεξάρτητα του μεγέθους N της λίστας)

Σχηματικά τα παραπάνω για κάθε περίπτωση έχουν ώς εξής





-Μέθοδος int get() αφαιρεί τον πρώτο κόμβο και επιστρέφει την τιμή του.Εάν η λίστα είναι άδεια,τότε προκύπτει NoSuchElementException γιατί δεν υπάρχει κεφαλη.Αλλιώς αφού η λίστα περιέχει πρώτο στοιχείο(head),κρατάμε την τιμή του(για να επιστραφεί στο τέλος) και εάν υπάρχει ένα στοιχείο μόνο στη λίστα(head==tail) τότε γίνονται null.Αλλιώς θα υπάρχουν >1 στοιχεία στην λίστα και πλέον κεφαλή θα είναι ο επόμενος κόμβος του head(πεδίο head.next).Σχηματικά έχουμε:



Ομοίως η αφαίρεση του στοιχείο ολοκληρώνεται σε O(1) πράξεις για τον ίδιο λόγο με τη μέθοδο put.

-Μέθοδος int size() επιστρέφει το μέγεθος της λίστας.Εάν η λίστα είναι άδεια προφανώς το μέγεθος που επιστρέφει είναι μηδέν.Αλλιώς,διασχίζοντας την λίστα από την κεφαλή μέχρι το στοιχείο που δείχνει σε null(τελευταίο στοιχείο) αυξάνουμε έναν μετρητή,ο οποίος επιστρέφεται στο τέλος.

-Μέθοδος void print(PrintStream stream) τυπώνει τα περιεχόμενα κάθε κόμβου.Η λίστα διασχίζεται από την κεφαλή μέχρι το τελευταίο στοιχείο(το οποίο θα δείχνει σε null).

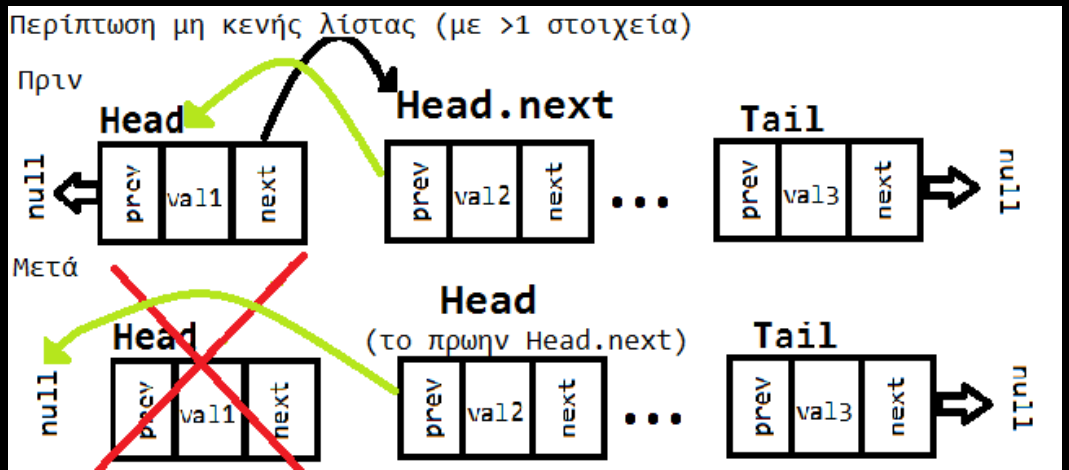
2)Τάξη DoubleEndedQueue(Παρατηρούμε ότι με την λίστα διπλής σύνδεσης είναι πολύ πιο έυκολη η αφαίρεση του τελευταίου στοιχείου σε σύγκριση με την λίστα μονής σύνδεσης)

-Private αναφορές head και tail που θα δείχνουν στο πρώτο και στο τελευταίο στοιχείο αντίστοιχα.

-Private inner class Node που περιλαμβάνει τα private μέλη int val(τιμή κόμβου),Node next και Node prev(αναφορά στον επόμενο και προηγούμενο κόμβο του τρέχοντος αντικειμένου Node).Ακόμη περιλαμβάνει τις κατάλληλες get μεθόδους για πρόσβαση στις παραπάνω private μεταβλητές.

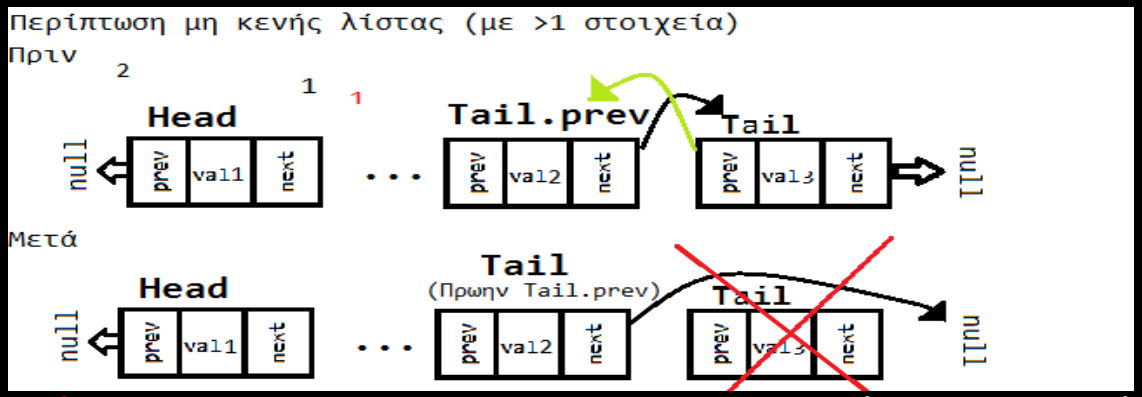
-Μέθοδοι getHead() , getTail() , isEmpty() ορίζονται ίδια με την IntQueueImpl

-Μέθοδοι int removeFirst()
υλοποιείται με το ίδιο σκεπτικό της get() της class IntQueueImpl με την διαφορά ότι πρέπει να ενημερώσουμε το πεδίο prev του επόμενου από την κεφαλή κόμβου.Σχηματικά η διαδικασία φαίνεται δίπλα.



Η μέθοδος εκτελείται σε O(1) πράξεις γιατί οι εντολές εκτελούνται λίγες φορές και είναι ανεξάρτητε του μεγέθους N της λίστας.

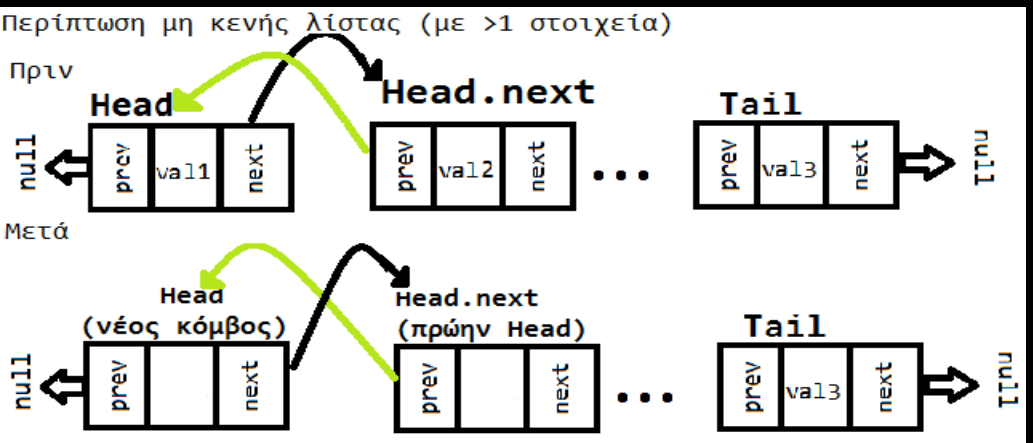
-Μέθοδος int removeLast() υλοποιείται με παρόμοιο σκεπτικό όπως η removeFirst αλλά προφανώς ενημερώνοντας τα κατάλληλα πεδία next και prev.Ομοίως εκτελείται σε O(1) για τους ίδιους λόγους.Σχηματικά έχουμε:



-Μέθοδοι getFirst() και getLast() που επιστρέφουν τις τιμές του πρώτου και του τελευταίου κόμβου αντίστοιχα

-Μέθοδος addFirst(int item) που τοποθετεί στην κεφαλή της λίστας ένα νέο κόμβο που περιέχει την τιμή item.Σχηματικά η μέθοδος κάνει τα ακόλουθα(εάν η λίστα δεν είναι κενή):

Εάν η λίστα είναι κενή τότε απλά ο νέος κόμβος είναι head και tail.



Τάξη IntQueueWithMinImpl

-Οι μέθοδοι isEmpty(),printQueue(PrintStream stream),size() δεν κάνουν τίποτα παραπάνω από το να καλούν τις ανάλογες μεθόδους των 2 προηγούμενων τάξεων.

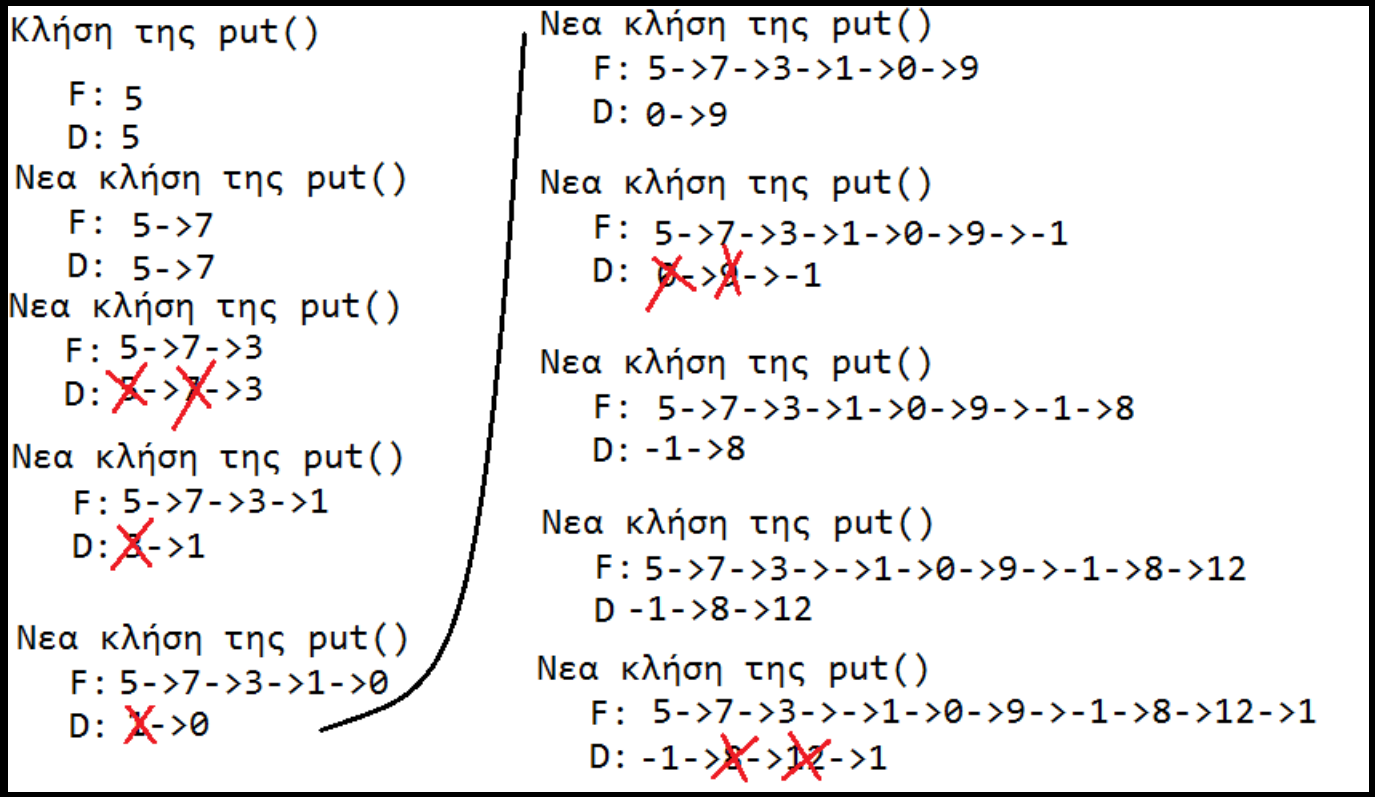
Επεξήγηση της υλοποίησης των μεθόδων put(int item),get() και min() που συντελούν στην επιστροφή του ελάχιστον στοιχείου της F σε O(1)

Ο αλγόριθμος για την put() είναι ο εξής:

Αρχικά τοποθετεί το int item στην F.Ακολούθως,έαν η D είναι άδεια το τοποθετεί και στην D.Εάν η D δεν είναι άδεια,τότε συγκρίνουμε το στοιχείο προς τοποθέτηση(item) με την κεφαλή της D.Εάν τιμη της κεφαλής>item τότε όσο η κεφαλή της D είναι μεγαλύτερη του item αφαιρείται η κεφαλή.Εάν μετά την αφαίρεση γίνει άδεια η D τότε ο βρόγχος σπάει.Αλλιώς συνεχίζει την εκτέλεση μέχρι να γίνει ψευδής η συνθήκη.Ακολούθως,προσθέτουμε το item με την addFirst στην D.Εαν όμως τιμή της κεφαλής <= item τότε απλά τοποθετούμε το στοιχείο στην D με την addFirst.Με αυτό τον τρόπο στην ουρά της D πάντα θα βρίσκεται το εκάστοτε minimum στοιχείο της F ακόμα και αν γίνει αφαίρεση του.

-Μέθοδος get() που αφαιρεί το παλιότερο στοιχείο της F θα αφαιρεί και από την D μόνο εάν το στοιχείο που αφαιρέθηκε από την F είναι ίσο με το τελευταίο στοιχείο της D(παλαιότερο).

Ακολουθεί ένα παράδειγμα εκτέλεσης της μεθόδου put()



-Μέθοδος min() με όσα προηγήθηκαν στην put() η min() υλοποιείται σε O(1) διότι απλά επιστρέφουμε την τιμή του tail της D.

Συμπερασματικά , η μέθοδος min() εκτελείται σε O(1) πράξεις , άρα ανεξάρτητος χρόνος εκτέλεσης από το πλήθος των δεδομένων.Εάν επιθυμούσαμε να υλοποιήσουμε την min() σαρώνοντας την εκάστοτε λίστα από την κεφαλή έως την ουρά ,τότε πάντα ο χρόνος θα ήταν O(N) αφού οι πράξεις θα εξαρτώνται από το πλήθος των στοιχείων της ουράς.Στο προηγούμενο παράδειγμα της put() ,συγκεκριμένα, θα έπρεπε να σαρώναμε την ουρά σε κάθε κλήση της (αλλά και σε κάθε κλήση της get()) για εύρεση ελάχιστου στοιχείου.