

# **PROJECT 1~ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ 2017**

ΟΜΑΔΑ : Γρηγοριάδου Γεωργία AM : 3160029

Γαλάτουλα Ναταλία AM : 3160019

## **ΜΕΡΟΣ Α-ΜΕΡΟΣ Γ**

- **Μέρος Α:**

Για την υλοποίηση της διεπαφής CharDoubleEndedQueue δημιουργήθηκε η κλάση doubleNode. Η κλάση doubleNode δημιουργεί κόμβους διπλής σύνδεσης οι οποίοι είναι τύπου character και συνδέονται μεταξύ τους με τα πεδία next και previous τα οποία δείχνουν στον επόμενο και στον προηγούμενο κόμβο αντίστοιχα.

Η υλοποίηση της διεπαφής CharDoubleEndedQueue επιτυγχάνεται στην κλάση CharDoubleEndedQueueImpl η οποία περιέχει έναν default constructor και 9 μεθόδους. Συγκεκριμένα υπάρχουν :

boolean isEmpty() :επιστρέφει true αν η λίστα είναι άδεια και false αν περιέχει τουλάχιστον έναν κόμβο.

Void addFirst(char item) : εισάγει κόμβους στην αρχή της λίστας. Με τη βοήθεια ενός κόμβου d δημιουργεί έναν κόμβο με όνομα head και στοιχείο το item. Ελέγχει αν η λίστα είναι άδεια, αν ισχύει τότε εισάγει τον κόμβο με όνομα head. Αν η λίστα περιέχει κόμβους τότε συνδέει τον καινούργιο κόμβο head με το βοηθητικό κόμβο d.Τέλος αυξάνει το μέγεθος της λίστας κατά 1.

char removeFirst() : διαγράφει κόμβους από την αρχή της λίστας με τη βοήθεια μιας μεταβλητής τύπου char η οποία κρατάει την τιμή του κόμβου head και με τη βοήθεια ενός κόμβου node1 ο οποίος κρατάει το πεδίο head.next. Ελέγχει αν η λίστα είναι άδεια, αν ισχύει τότε πετάει new NoSuchElementException. Αν η λίστα περιέχει έναν κόμβο τον κάνει null αλλιώς διαγράφει τον πρώτο κόμβο της λίστας.Τέλος αφαιρεί το μέγεθος της λίστας κατά 1.

Void addLast(char item): ίδια με την addFirst όμως η addLast εισάγει τους κόμβους στο τέλος της λίστας.

Char removeLast(): ίδια με την removeFirst όμως η removeLast διαγράφει κόμβους από το τέλος της λίστας.

Char getLast και char getFirst : επιστρέφουν το στοιχείο του τελευταίου και του πρώτου κόμβου αντίστοιχα. Αν η λίστα είναι άδεια τότε πετάει new NoSuchElementException().

Int size() : επιστρέφει το μέγεθος της λίστας.

printQueue(PrintStream stream) : εμφανίζει τα στοιχεία της λίστας.

- **Μέρος Γ:**

Για την υλοποίηση της διεπαφής CharQueue δημιουργήθηκε η κλάση Node. Η κλάση Node δημιουργεί κόμβους απλής σύνδεσης οι οποίοι είναι τύπου character και συνδέονται μεταξύ τους με το πεδίο next το οποίο δείχνει στον επόμενο κόμβο.

Η υλοποίηση της διεπαφής CharQueue επιτυγχάνεται στην κλάση CharQueueImpl η οποία περιέχει έναν default constructor και 6 μεθόδους. Συγκεκριμένα υπάρχουν :

boolean isEmpty() :επιστρέφει true αν η ουρά είναι άδεια και false αν περιέχει τουλάχιστον έναν κόμβο.

Void put(char item) : εισάγει κόμβους στην ουρά με τη με τη μέθοδο first in first out,δηλαδή εισάγει τους κόμβους τον έναν μετά τον άλλο.

Char get(): διαγράφει τον κόμβο που είναι εισαχθεί πρώτο απ' όλους τους κόμβους στην ουρά και τον επιστρέφει. Όμως πρώτα απ' όλα ελέγχει αν η ουρά είναι άδεια, αν ισχύει τότε πετάει new NoSuchElementException().

Char peek(): Ελέγχει αν η ουρά είναι άδεια και έπειτα επιστρέφει τον πρώτο κόμβο.

Void printQueue(): εμφανίζει τους κόμβους της ουράς.

Int size(): επιστρέφει το μέγεθος της ουράς.

## **ΜΕΡΟΣ Β**

Η εφαρμογή DNAPalindrome δέχεται μια ακολουθία κεφαλαίων χαρακτήρων και ελέγχει εάν είναι Watson-Crick complemented palindrome. Η υλοποίηση της εφαρμογής με τη χρήση των μεταβλητών int size, String dna και boolean bool η οποία καθορίζεται εξ αρχής true. Αρχικά δημιουργείται ένα αντικείμενο τύπου CharDoubleEndedQueueImpl δηλαδή μια λίστα διπλής σύνδεσης. Ο χρήστης πληκτολογεί την ακολουθία χαρακτήρων η οποία αποθηκεύεται στο String dna. Μετά αποθηκεύεται το μέγεθος του String στην μεταβλητή size με την εντολή dna.length(). Στο πρόγραμμα υπάρχει εντολή η οποία δεν επιτρέπει στο χρήστη να εισάγει κάτι διαφορετικό από κεφαλαίους χαρακτήρες και το κενό string. Η εντολή είναι η dna.matches("[A-Z ] +"). Στη συνέχεια με μία for εισάγονται οι χαρακτήρες ένας προς έναν στη λίστα με τη βοήθεια της εντολής charAt(i) η οποία σπάει το String σε χαρακτήρες. Έπειτα μέσω της λίστας και της εντολής της getFirst() και getLast() γίνεται ο έλεγχος για το ζητούμενο αποτέλεσμα. Αν το αποτέλεσμα είναι διαφορετικό από αυτό που ζητείται τότε η μεταβλητή bool γίνεται false. Τέλος αν η bool είναι true το πρόγραμμα εμφανίζει ότι η ακολουθία χαρακτήρων είναι Watson-Crick complemented palindrome αλλιώς εμφανίζει ότι δεν είναι.

## **ΜΕΡΟΣ Δ**

Η κλάση CharQueueWithMinImpl υλοποιεί τις μεθόδους της διεπαφής CharQueueWithMin. Η κλάση αποτελείται από τις ίδιες μεθόδους της κλάσης CharQueueImpl με μία επιπλέον μέθοδο την char min() η οποία επιστρέφει το μικρότερο χαρακτήρα στην ουρά. Αρχικά στην κλάση δημιουργούνται μία ουρά F της κλάσης CharQueueImpl και μία διπλά συνδεδεμένη λίστα D της κλάσης CharDoubleEndedQueueImpl. Με αυτά τα δύο αντικείμενα παρέχεται πρόσβαση στις μεθόδους των παραπάνω κλάσεων και με τη χρήση τους υλοποιούνται και οι μέθοδοι της κλάσης CharQueueWithMinImpl. Έπειτα δημιουργείται ένας default constructor για τη δημιουργία αντικειμένων κλάσης CharQueueWithMinImpl. Η μέθοδος void put(char item) εισάγει το στοιχείο item στην ουρά F με την εντολή F.put(item) και στη συνέχεια ελέγχει αν η λίστα D είναι άδεια, αν ισχύει τότε εισάγει το item πρώτο στη λίστα. Όταν ο χρήστης θα εισάγει δεύτερο χαρακτήρα τότε η μέθοδος put θα το εισάγει πρώτα στην ουρά και μετά στη λίστα όμως αυτή τη φορά θα υπάρχει ήδη ένα item οπότε η put θα πάρει το τελευταίο item της λίστας το οποίο είχε εισαχθεί προηγουμένως και θα ελέγξει αν είναι μικρότερο του στοιχείου που εισήγαγε δεύτερο ο χρήστης. Κάθε φορά θα συγκρίνεται το επόμενο item με το προηγούμενο. Όταν το item είναι μεγαλύτερο το προηγούμενο εισάγεται πρώτο στη λίστα ενώ αν είναι μικρότερο εισάγεται τελευταίο. Έτσι επιτυγχάνεται ο μικρότερος χαρακτήρας να είναι πάντα στο τέλος της λίστας. Με αυτό τον τρόπο η μέθοδος char min() επιστρέφει το μικρότερο item σε χρόνο O(1) επιστρέφοντας το τελευταίο item της λίστας. Στη συνέχεια η μέθοδος char get() επιστρέφει και αποθηκεύει σε μία μεταβλητή char c το παλαιότερο στοιχείο της ουράς F με την εντολή F.peek() και μετά το διαγράφει με την εντολή F.get(), έπειτα ελέγχει αν η λίστα D είναι άδεια, αν ισχύει τότε πετάει new NoSuchElementException(), αν όχι τότε διαγράφει το πρώτο στοιχείο που εισήχθη στη λίστα. Η μέθοδος char peek() ελέγχει αν η ουρά F είναι άδεια, αν ισχύει τότε πετάει

`new NoSuchElementException()`, αν όχι επιστρέφει το παλαιότερο item που βρίσκεται στην ουρά. Η μέθοδος `int size()` επιστρέφει το μέγεθος της ουράς `F` και η `void printQueue` εμφανίζει όλα τα στοιχεία της.