

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΕΡΓΑΣΙΑ 1

P3220066 Αριστείδης Καραγιαννάκος

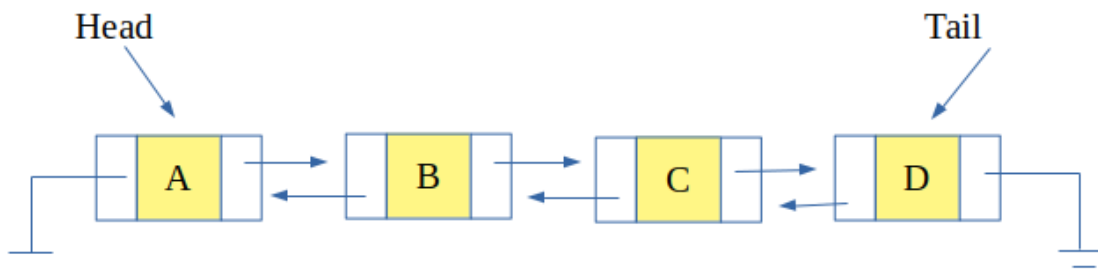
P3220037 Έλλη-Μαρία Δανέζη

Θέμα Α

Interface StringDoubleEndedQueue: Υλοποιούμε το interface στην StringDoubleEndedQueueImpl κλάση.

Ακόμη κάνουμε χρήση generics έτσι ώστε να μπορούμε να δώσουμε ό,τι αντικείμενο θέλουμε.

Το interface είναι ένα προσχέδιο μιας κλάσης java που περιέχει αφηρημένες μεθόδους και σταθερές ιδιότητες. Η διεπαφή περιλαμβάνει τη δήλωση μεθόδων και αυτές οι μέθοδοι θα οριστούν στην κλάση.



- Οι κλασεις στο StringDoubleEndedQueueImpl είναι σε T data:
 1. isEmpty(): ελέγχει αν η λίστα είναι άδεια και επιστρέφει True ή False.

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΕΡΓΑΣΙΑ 1

P3220066 Αριστείδης Καραγιαννάκος

P3220037 Έλλη-Μαρία Δανέζη

2. `addFirst()`: εισάγει στην αρχή της λίστας ένα τύπου `T` αντικείμενο σε $O(1)$ χρόνο, δηλαδή σε σταθερό χρόνο, αφού υλοποιείται ανεξάρτητα του `N` (μέσω των δεικτών `head`, `tail`, `next`, `prev`) Αφου γινουν η εντολες αλλαζουμε και τους δεικτες
3. `removeFirst()`: διαγράφει το πρώτο αντικείμενο που έχει εισαχθεί στην λίστα $O(1)$ χρόνο για τον ίδιο ακριβώς λόγο με το `addFirst ()`.
4. `addLast()`: εισάγει στο τέλος τις λίστας ένα αντικείμενο τύπου `T` σε $O(1)$ χρόνο, για τον ίδιο ακριβώς λόγο με το `addFirst ()`.
5. `removeLast()`: διαγράφει από το τέλος τη λίστας ς ένα τύπου `T` αντικείμενο σε $O(1)$ χρόνο για τον ίδιο ακριβώς λόγο με το `addFirst ()`.
6. `getFirst()`: παίρνει το πρώτο `T` αντικείμενο από την λίστα.
7. `getLast()`: Παίρνει το τελευταίο `T` αντικείμενο από τη λίστα .
8. `printQueue()`: αναλόγως με το όρισμα που δίνεται κάνει και την εντολή. Πχ αν το `stream= System.out` εκτυπώνει τη λίστα.
9. `size()`: Δείχνει το μέγεθος της λίστας. Έχουμε ορίσει μια μεταβλητή `count` και κάθε φορά που προστίθεται ένα στοιχείο στην λίστα, αυξάνουμε τον `count` κατά 1, ενώ κάθε φορά που αφαιρείται ένα στοιχείο από τη λίστα,

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΕΡΓΑΣΙΑ 1

P3220066 Αριστείδης Καραγιαννάκος

P3220037 Έλλη-Μαρία Δανέζη

μειώνουμε τον count κατά 1. Η size () επιστρέφει την μεταβλητή count, κάτι που γίνεται σε σταθερό χρόνο, οπότε η size λειτουργεί σε $O(1)$ χρόνο.

Θεμα Β:

Στο θεμα Β χρησιμοποιήσαμε την κλάση από το θέμα Α, «StringDoubleEndedQueueImpl» για να κάνουμε τη μετατροπή από προθεματική (prefix) μορφή σε ενθεματική (infix) μορφή.

ariumitika_prajeis: Ελέγχει αν αυτό που του δίνεται είναι αριθμός ή τελεστής και επιστρέφει True (αν είναι τελεστής) αλλιώς False.

an ariumtoi: Ελέγχει αν αυτό που του δίνεται είναι αριθμός ή τελεστής και επιστρέφει True αν είναι αριθμός αλλιώς False .

convert: δεχεται είναι String αν ένα από αυτά είναι True από την ariumitika_prajeis τότε κανει remove τα δυο στοιχεία που έχει η λιστα από το θεμα α και τα προσθετει με το συμβολο και παρενθεσεις για πρωτεραιότητα πραξεων αλλιως αν ariumitika_prajeis είναι False τότε προσθετει στην λιστα το σύμβολο που δινεται από το String στο return απλως δινει το τελευταίο στοιχείο τις λιστας . αυτά υλοποιουνται από της μεθόδους addlast και removelast από το θεμαςΑ.

Στο τέλος της main διαβάζει τα χαρακτηριστικά ενός prefix to infix και δείχνει αν είναι σωστό αυτό που έχει καταχωρηθεί με τις εξείς περιπτώσεις.

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΕΡΓΑΣΙΑ 1

P3220066 Αριστείδης Καραγιαννάκος

P3220037 Έλλη-Μαρία Δανέζη

Το String που θα εισαχθεί θα πρέπει να έχει παραπάνω αριθμούς από τους τελεστές πιο συγκεκριμένα θα πρέπει να είναι κατά ένα παραπάνω σε count αριθμοί.

Αυτό που θα εισαχθεί θα πρέπει να είναι μόνο αριθμοί από το 0-9 και σύμβολα +-*/.

Τέλος αυτό που θα εισαχθεί τα τελευταία δύο ψηφία θα πρέπει να είναι αριθμοί και αν είναι τότε το count των τελεστών θα πρέπει να είναι μεγαλύτερο από το count των αριθμών.

Πώς υλοποιείται βήμα βήμα:

1. Διαβάζουμε το String
2. Αν είναι ψηφίο το τοποθετούμε στην λίστα
3. Εάν είναι τελεστής κάνουμε 2 στοιχεία από την λίστα remove και εφαρμόζουμε την πράξη σαν String
4. Επανάληψη μέχρι τέλους String.

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΕΡΓΑΣΙΑ 1

P3220066 Αριστείδης Καραγιαννάκος

P3220037 Έλλη-Μαρία Δανέζη

Θεμα Γ:

Το θέμα Γ αποτελείται από ακολουθιακή δομή, μία δομή επανάληψης και τέσσερις δομές επιλογής.

Συγκεκριμένα, η κλάση `DNAPalindrome` περιέχει τη μέθοδο `complem()` – δέχεται δύο μεταβλητές τύπου χαρακτήρα και επιστρέφει τιμή `Boolean` – η οποία ελέγχει αν οι δύο μεταβλητές που παίρνει σαν όρισμα είναι συμπληρωματικές. Δηλαδή, αν είναι συνδυασμός A-T ή C-G και αντίστροφα.

Στην `main class` ο χρήστης καλείται να εισάγει σε σωστή μορφή την ακολουθία γενετικού κώδικα σε μορφή `String` και ύστερα γίνεται μετατροπή αυτού του τύπου σε τύπο χαρακτήρα ώστε να εισαχθεί στην ουρά από το θέμα α και να γίνει η σύγκριση των χαρακτήρων μέσω της μεθόδου που προαναφέρθηκε. Αν το DNA είναι συμμετρικά συμπληρωματικό, δηλαδή αν συγκρίνοντας τον πρώτο χαρακτήρα του `String` με τον τελευταίο και κάθε φορά τον επόμενο του πρώτου με τον προηγούμενο του τελευταίου και όλα είναι συμπληρωματικά τότε το `flag` παραμένει 1 και εμφανίζεται στην οθόνη ότι η ακολουθία γενετικού κώδικα είναι συμπληρωματικά παλίνδρομη, διαφορετικά το `flag` γίνεται 0 και εμφανίζεται το αντίστοιχο μήνυμα στην οθόνη.

Τα παραπάνω γίνονται σε $O(N)$ χρόνο καθώς υλοποιούνται με τη χρήση μίας επανάληψης που εξαρτάται από το πλήθος N των στοιχείων του DNA.

Παραδειγμα: ATAT

Watson-crick complement: TATA