

Μέρος Α – Υλοποίηση των Διεπαφών StringStack και DoubleQueue

StringStackImpl

Η κλάση StringStackImpl υλοποιεί τη διεπαφή StringStack και βασίζεται σε λίστα μονής σύνδεσης. Οι βασικές λειτουργίες που υποστηρίζει είναι οι εξής:

push(String item):

Προσθέτει ένα νέο στοιχείο στην κορυφή της στοίβας.

1. Δημιουργούμε ένα καινούριο Node που περιέχει το δεδομένο item.
2. Θέτουμε το επόμενο του newNode να είναι το τρέχον top της κλάσης (με χρήση του this.top).
3. Αυτό εξασφαλίζει ότι το νέο στοιχείο συνδέεται. Ενημερώνουμε το top να δείχνει στο newNode

pop():

Αφαιρεί και επιστρέφει το στοιχείο από την κορυφή της στοίβας. Αν η στοίβα είναι κενή, εκτοξεύεται η εξαίρεση NoSuchElementException.

1. Ελέγχουμε αν η στοίβα είναι κενή με τη μέθοδο isEmpty().
2. Ενημερώνουμε το top ώστε να δείχνει στο επόμενο στοιχείο της στοίβας
3. Ενημερώνουμε το size
4. Επιστρέφουμε το αποθηκευμένο δεδομένο που αφαιρέθηκε.

isEmpty():

Ελέγχει αν η στοίβα είναι κενή.

size():

Επιστρέφει το πλήθος των στοιχείων της στοίβας.

DoubleQueueImpl

Η κλάση DoubleQueueImpl υλοποιεί τη διεπαφή DoubleQueue και χρησιμοποιεί λίστα μονής σύνδεσης.

enqueue(double item):

Προσθέτει ένα νέο στοιχείο στο τέλος της ουράς.

1. Δημιουργούμε ένα καινούριο Node που περιέχει το δεδομένο item.
2. Ελέγχουμε αν η ουρά είναι κενή. Αν είναι, θέτουμε το head να δείχνει στο newNode.
3. Διαφορετικά, θέτουμε το next του τελευταίου στοιχείου της ουράς να δείχνει στο newNode.
4. Ενημερώνουμε το tail ώστε να δείχνει στο newNode.
5. Αυξάνουμε το size της ουράς κατά 1.

dequeue():

Αφαιρεί και επιστρέφει το στοιχείο από την αρχή της ουράς. Αν η ουρά είναι κενή, πετάει την εξαίρεση NoSuchElementException.

1. Ελέγχουμε αν η ουρά είναι κενή με τη μέθοδο isEmpty().
2. Αποθηκεύουμε το δεδομένο του πρώτου στοιχείου (head.data).
3. Ενημερώνουμε το head ώστε να δείχνει στο επόμενο στοιχείο της ουράς (head.next).
4. Αν το head είναι πλέον null, θέτουμε και το tail σε null.
5. Μειώνουμε το size κατά 1.
6. Επιστρέφουμε το αποθηκευμένο δεδομένο.

peek():

Επιστρέφει το πρώτο στοιχείο της ουράς χωρίς να το αφαιρεί.

isEmpty():

Ελέγχει αν η ουρά είναι κενή.

size():

Επιστρέφει το πλήθος των στοιχείων της ουράς.

printQueue():

Τυπώνει όλα τα στοιχεία της ουράς με τη σειρά που υπάρχουν

Μέρος Β – Έλεγχος Ταίριασματος Ετικετών σε HTML

Στο Μέρος Β, υλοποιήθηκε το πρόγραμμα TagChecking.java, το οποίο ελέγχει αν οι ετικέτες (tags) ενός HTML αρχείου είναι σωστά ταιριασμένες

Χρήση της StringStackImpl

Η υλοποίηση της StringStackImpl παρέχει βασικές λειτουργίες στοίβας (push, pop, peek και isEmpty) με χρόνο $O(1)$. Αυτές οι λειτουργίες είναι ιδανικές για τον έλεγχο αντιστοιχίας ετικετών, καθώς η στοίβα λειτουργεί με τη λογική Last In, First Out (LIFO), η οποία απαιτείται για την επεξεργασία εμφωλευμένων ετικετών.

1. push:

Χρησιμοποιείται για την προσθήκη ετικετών ανοίγματος στη στοίβα.

2. pop:

Χρησιμοποιείται για την αφαίρεση της τελευταίας ετικέτας από τη στοίβα όταν βρίσκεται το αντίστοιχο κλείσιμο.

3. peek:

Χρησιμοποιείται για να ελέγξουμε αν η τρέχουσα ετικέτα κλεισίματος ταιριάζει με την κορυφαία ετικέτα της στοίβας.

4. isEmpty:

Χρησιμοποιείται για να ελέγξουμε αν υπάρχουν μη κλειστές ετικέτες στο τέλος του αρχείου.

Περιγραφή της Λειτουργίας του Προγράμματος

1. Ανάγνωση Εισόδου:

- Το όνομα του αρχείου δίνεται είτε μέσω παραμέτρου εντολών (args[0]) είτε εισάγεται από τον χρήστη κατά την εκτέλεση.
- Χρησιμοποιείται BufferedReader για τη γραμμή-γραμμή ανάγνωση του αρχείου.

2. Επεξεργασία Γραμμών:

- Κάθε γραμμή καθαρίζεται από περιττά κενά και ελέγχεται για την παρουσία ετικετών με τα σύμβολα < και >.
- Με χρήση substring εντοπίζονται οι ετικέτες, και διαχωρίζουμε:
- Ετικέτες ανοίγματος: Προστίθενται στη στοίβα με τη push.
- Ετικέτες κλεισίματος:
 - i. Ελέγχεται αν η στοίβα είναι κενή ή αν η κορυφαία ετικέτα δεν ταιριάζει με αυτή που κλείνει.
 - ii. Σε περίπτωση μη ταύτισης, εκτυπώνεται μήνυμα σφάλματος και το πρόγραμμα τερματίζεται.
 - iii. Αν ταιριάζουν, αφαιρείται η κορυφαία ετικέτα με τη pop.

3. Τελικός Έλεγχος:

- Αν στο τέλος της ανάγνωσης η στοίβα είναι κενή, τότε όλες οι ετικέτες έχουν ταίριασμα.
- Διαφορετικά, εκτυπώνεται μήνυμα ότι υπάρχουν μη κλειστές ετικέτες

Μέρος Γ – Εύρεση καθαρού κέρδους από την πώληση μετοχών

Στο Μέρος Γ, υλοποιήθηκε το πρόγραμμα NetProfit.java οποίο υπολογίζει το καθαρό κέρδος/ζημία από την αγοροπώληση μετόχων.

Χρήση της DoubleQueueImpl

Η υλοποίηση της DoubleQueueImpl παρέχει τις βασικές λειτουργίες ούρας(enqueue, dequeue, peek, isEmpty) με χρόνο $O(1)$. Αυτές οι λειτουργίες είναι ιδανικές για την πώληση τον υπολογισμό καθαρού κέρδους/ζημίας και ειδικότερα των παλαιότερων μετοχών, καθώς η ουρά χρησιμοποιεί τη λογική First-In-First-Out(FIFO), η οποία απαιτείται για την πώληση των παλαιότερων μετοχών.

1.enqueue: Χρησιμοποιείται για την εισαγωγή της τιμής και ποσότητας μετοχών στο πίσω μέρος της ουράς

2.dequeue: Χρησιμοποιείται για την εξαγωγή της τιμής και ποσότητας μετοχών από το μπροστινό μέρος της ουράς

Περιγραφή της λειτουργίας του Προγράμματος

1.Ανάγνωση Εισόδου

-Το όνομα του αρχείου txt δίνεται είτε μέσω παραμέτρου εντολών(args[0]) είτε εισάγεται από τον χρήστη κατά την εκτέλεση.

-Χρησιμοποιείται BufferedReader για την γραμμή προς γραμμή ανάγνωση του αρχείου.

2.Λειτουργία αγοράς μετοχών

-Κάθε γραμμή χωρίζεται σε ένα string array parts ανάλογα με τα whitespaces, δηλαδή αν η γραμμή είναι buy 10 price 15,5 τότε parts[0]=buy, parts[1]=10 etc.

-Όταν το parts[0] είναι buy τότε εισάγουμε στην ούρα την τιμή και την ποσότητα των μετοχών που αγοράστηκαν καθώς και την ποσότητα των μετοχών που αγοράστηκαν στο σύνολο των μετοχών totalStocks

3.Λειτουργία πώλησης μετοχών

-Όταν το parts[0] είναι sell τότε ελέγχουμε ότι η ποσότητα των μετοχών για πώληση είναι μικρότερο από το σύνολο των μετοχών που διαθέτουμε και ότι η ποσότητα των μετοχών που θα πουληθεί είναι θετική.

-Αν είναι θετική, τότε γίνεται εξαγωγή της τιμής και ποσότητας που αγοράστηκαν πρώτα και βρίσκονται στο μπροστινό μέρος της ουράς και γίνεται σύγκριση των ποσοτήτων με την χρήση του Math.min για να δούμε αν χρειάζεται μετά άλλη εξαγωγή από το μπροστινό μέρος της ουράς, καθώς και για να ανανεωθεί η ποσότητα των μετοχών που πρέπει να πωληθούν ακόμα αν η ποσότητα πώλησης είναι μεγαλύτερη από την ποσότητα αγοράς που βρίσκεται στο μπροστινό μέρος της ουράς.

- Ταυτόχρονα, ανανεώνεται ο συνολικός αριθμός μετοχών totalStocks και όποια από τις δύο ποσότητες είναι μικρότερη εκείνη αφαιρείται από τον αριθμό των μετοχών που πρέπει να πωληθούν σε αυτή τη γραμμή
- Τέλος, αν η ποσότητα μετοχών που πρέπει να πωληθεί είναι μικρότερη από την ποσότητα των μετοχών στο μπροστινό μέρος της ουράς τότε η μετοχές που περισσεύουν επανατοποθετούνται στην ουρά με την ανανεωμένη ποσότητα. Την ίδια στιγμή υπολογίζεται το NetProfit, το οποίο είναι το καθαρό κέρδος.

Μελη-

Σπύρος Παναγιώτης Σκούρας 3190185

Θεόδωρος Μανδαλάκης 3230115