

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

1Η ΕΡΓΑΣΙΑ

ΜΕΡΟΣ Α΄

StringStackImpl:

Στην StringStackImpl υλοποιούνται οι βασικές λειτουργίες μιας στοίβας.

Οι μέθοδοι που χρησιμοποιούνται είναι οι εξής:

IsEmpty: Η μέθοδος IsEmpty ελέγχει και επιστρέφει Boolean αποτέλεσμα για το αν η λίστα είναι κενή ή όχι.

Push: Η μέθοδος push δέχεται ως όρισμα ένα στοιχείο και δημιουργεί ένα αντικείμενο λίστας καλώντας την ListNode. Στη συνέχεια ελέγχει αν η λίστα είναι κενή χρησιμοποιώντας την μέθοδο IsEmpty. Αν είναι αποθηκεύει το στοιχείο ως head και ως tail, αν όχι το αποθηκεύει ως tail

Pop: Η μέθοδος pop αρχικά ελέγχει αν η λίστα είναι άδεια χρησιμοποιώντας την μέθοδο IsEmpty. Αν είναι τότε καλεί την NoSuchElementException. Αν όχι παίρνει το head στοιχείο και το διαγράφει. Στη συνέχεια αποθηκεύει ως head στοιχείο το αμέσως προηγούμενο και το επιστρέφει.

Peek: Η μέθοδος peek ελέγχει αν η λίστα είναι κενή χρησιμοποιώντας την μέθοδο IsEmpty. Αν είναι καλεί την NoSuchElementException, ενώ αν δεν είναι επιστρέφει το head στοιχείο της λίστας.

PrintStack: : Η μέθοδος PrintStack αρχικά ελέγχει αν η λίστα είναι άδεια χρησιμοποιώντας την μέθοδο IsEmpty. Αν είναι τότε καλεί την NoSuchElementException, ενώ αν δεν είναι, εκτυπώνει τα στοιχεία της λίστας από το τελευταίο στο πρώτο χρησιμοποιώντας μια δομή while και καλώντας κάθε φορά το επόμενο στοιχείο στη λίστα μέχρι να μην υπάρχει επόμενο στοιχείο.

Size: Η μέθοδος size αρχικοποιεί ένα μετρητή στο μηδέν και τον αυξάνει διασχίζοντας την λίστα μέχρι να φτάσει στο πρώτο στοιχείο χρησιμοποιώντας μια δομή while.

IntQueueImpl:

Η κλάση IntQueueImpl υλοποιεί τις βασικές λειτουργίες μιας ουράς.

Οι μέθοδοι που χρησιμοποιούνται είναι οι εξής:

IsEmpty: Η μέθοδος IsEmpty ελέγχει και επιστρέφει Boolean αποτέλεσμα για το αν η λίστα είναι κενή ή όχι.

Put: Η μέθοδος put δεχεται ως όρισμα ένα στοιχείο και δημιουργεί με αυτό ένα στοιχείο λίστας καλώντας την ListNode. Στη συνέχεια ελέγχει αν η λίστα είναι κενή χρησιμοποιώντας την μέθοδο IsEmpty. Αν είναι αποθηκεύει το στοιχείο ως head και ως tail, αν όχι το αποθηκεύει ως tail.

Get: Η μέθοδος get αρχικά ελέγχει αν η λίστα είναι άδεια χρησιμοποιώντας την μέθοδο IsEmpty. Αν είναι τότε καλεί την NoSuchElementException. Αν όχι παίρνει το head στοιχείο και το διαγράφει. Στη συνέχεια αποθηκεύει ως head στοιχείο το αμέσως προηγούμενο και το επιστρέφει.

Peek: Η μέθοδος peek ελέγχει αν η λίστα είναι κενή χρησιμοποιώντας την μέθοδο IsEmpty. Αν είναι καλεί την NoSuchElementException, ενώ αν δεν είναι επιστρέφει το head στοιχείο της λίστας.

PrintQueue: : Η μέθοδος PrintQueue αρχικά ελέγχει αν η λίστα είναι άδεια χρησιμοποιώντας την μέθοδο IsEmpty. Αν είναι τότε καλεί την NoSuchElementException, ενώ αν δεν είναι, εκτυπώνει τα στοιχεία της λίστας από το πρώτο στο τελευταίο χρησιμοποιώντας μια δομή while και καλώντας κάθε φορά το επόμενο στοιχείο στη λίστα μέχρι να μην υπάρχει επόμενο στοιχείο.

Size: Η μέθοδος size αρχικοποιεί ένα μετρητή στο μηδέν και τον αυξάνει διασχίζοντας την λίστα μέχρι να φτάσει στο τελευταίο στοιχείο χρησιμοποιώντας μια δομή while.

ΜΕΡΟΣ Β΄

Στο μέρος β΄ διαβάζονται ένα προς ένα τα html tags μέσα από ένα αρχείο html χρησιμοποιώντας bufferedReader. Καθώς διαβάζονται γίνεται έλεγχος αν τα tags είναι opening tags ή closing tag καλώντας την μέθοδο matcher. Τα opening tags τα οποία διαβάζονται, αποθηκεύονται στη στοίβα. Όταν διαβάζεται ένα closing tag γίνεται έλεγχος αν το συγκεκριμένο closing tag ταιριάζει με το head στοιχείο χρησιμοποιώντας την μέθοδο peek. Αν τα tags ταιριάζουν τότε καλείται η μέθοδος pop και στη συνέχεια διαβάζει το επόμενο tag. Αν τα tags δεν ταιριάζουν, τότε το πρόγραμμα εκτυπώνει μήνυμα για μη ταιριασμένα tags και σταματάει να τρέχει. Όταν διαβαστούν όλα τα tags γίνεται έλεγχος για το αν η λίστα είναι κενή καλώντας την μέθοδο IsEmpty και αν είναι, τότε εκτυπώνεται μήνυμα ότι τα tags είναι ταιριασμένα, αλλιώς εκτυπώνεται μήνυμα ότι τα tags δεν είναι ταιριασμένα (Ουσιαστικά μόνο στην περίπτωση που δεν είναι ταιριασμένο το πρώτο tag που έχει διαβαστεί).

ΜΕΡΟΣ Γ΄

Στο μέρος γ' διαβάζεται ανα μία γραμμή ένα .txt αρχείο το οποίο στην συγκεκριμένη μορφή που μας έχει δωθεί μπορεί να πραγματοποιήσει λογιστικές συναλλαγές τύπου buy και sell και στο τέλος να τυπώσει αν έχουμε συνολικό κέρδος(Total Profit) ή ζημία(Total Damage), το οποίο γινετε χρησιμοποιώντας BufferedReader. Καθώς διαβάζεται το αρχείο υπολογίζεται το καθαρό κέρδος μέσω μιας ουράς FIFO και αναλογα την συναλλαγή κάνει put (buy) ή get(sell) Σε περίπτωση που ζητηθούν παραπάνω μετοχές προς πώληση από αυτές που ήδη έχουμε τυπώνεται ανάλογο μήνυμα αναφέροντας επίσης την ποσότητα των μετοχών που έχουμε και αυτής που μας ζητήθηκε να πουλήσουμε (μετά το πρόγραμμα τερματίζεται). Σημαντικό ρολο στην υλοποίηση της μεθόδου είχε η μέθοδος split η οποία μας βοήθησε να σπάσουμε σε κομμάτια τα Strings της κάθε γραμμής του .txt αρχείου.