Θέμα Α

Κλάση Node

Η κλάση Node αναπαριστά τον κόμβο μίας μονά συνδεδεμένης λίστας. Τα μέλη της είναι το data αντιστοιχεί στο στοιχείο το οποίο είναι αποθηκευμένο στον κόμβο και το head που είναι δείκτης στον επόμενο κόμβο.

Ο κατασκευαστής Node(T data) αρχικοποιεί την τιμή του μέλους data.

Η μέθοδος setNext(Node<T> my_node) δίνει την τιμή my_node στο μέλος next του κόμβου, δηλαδή θέτει ως επόμενο κόμβο του τρέχοντος κόμβου τον κόμβο my_node.

Η μέθοδος Node<T> getNext() επιστρέφει τον επόμενο κόμβο.

Η μέθοδος setData(T data) θέτει την τιμή του μέλους του πεδίου data της κλάσης ίση με το όρισμα της μεθόδου.

Η μέθοδος T getData() επιστρέφει το data του κόμβου

Κλάση List

Η κλάση List αναπαριστά μία μονά συνδεδεμένη λίστα. Τα πεδία της το head που αποτελεί δείκτη στον πρώτο κόμβο της λίστας και το tail που είναι δείκτης στον τελευταίο κόμβο της λίστας.

Η μέθοδος isEmpty() επιστρέφει true αν η λίστα είναι άδεια και false αν δεν είναι άδεια.

Η μέθοδος insertAtFront(Node<T> my_node) εισάγει τον κόμβο my_node στην αρχή της λίστας. Αν η λίστα είναι κενή τότε θέτει την τιμή των δεικτών head και tail ίση με το my_node.Διαφορετικά, θέτει ως επόμενο κόμβο του κόμβου my_node τον κόμβο στον οποίο δείχνει ο δείκτης head και στη συνέχεια δίνει την τιμή my_node στον κόμβο head.

Η μέθοδος removeFromFront() αφαιρεί και επιστρέφει το data του πρώτου κόμβο της λίστας. Αν η λίστα είναι κενή τότε πετάει μία εξαίρεση NoSuchElementException. Σε διαφορετική περίπτωση, αν υπάρχει μόνο ένας κόμβος στη λίστα ,τότε θέτουμε το head και το tail ίσο με null και επιστρέφουμε το data αλλιώς θέτουμε το head ίσο με τον επόμενο κόμβο στον οποίο δείχνει το head και επιστρέφουμε το data.

Η μέθοδος insertAtEnd(Node<T> my_node) εισάγει έναν κόμβο στο τέλος της λίστας. Αν η λίστα είναι κενή ,τότε θέτουμε τους δείκτες head και tail ίσους με το my_node ,αλλιώς θέτουμε τον επόμενο κόμβο του tail ίσο με το my_node και τον δείκτη tail ίσο με τον my_node.

Η μέθοδος removeFromEnd() αφαιρεί τον τελευταίο κόμβο και επιστρέφει το data του. Αν η λίστα είναι κενή ,τότε ρίχνει εξαίρεση NoSuchElementException αλλιώς αν υπάρχει μόνο ένας κόμβος λίστα θέτει τους δείκτες head και tail ίσους με το null και επιστρέφει το data.Διαφορετικά, ψάχνουμε να βρούμε τον προτελευταίο κόμβο και θέτουμε τον επόμενό του ίσο με null.Έπειτα, θέτουμε το tail ίσο με τον προτελευταίο κόμβο και επιστρέφει το data.

Η μέθοδος returnFromFront() επιστρέφει χωρίς να αφαιρεί το data του τελευταίου κόμβου .Αν η λίστα είναι κενή ρίχνει μια εξαίρεση NoSuchElementException .Σε διαφορετική περίπτωση, επιστρέφει το data του tail.

Η μέθοδος returnFromFront() επιστρέφει χωρίς να αφαιρεί το data του πρώτου κόμβου.Αν η λίστα είναι κενή ,τότε ρίχνει μία εξαίρεση NoSuchElementException αλλιώς επιστρέφει το data του head.

Η μέθοδος print(PrintStream printStream) εκτυπώνει τα στοιχεία της λίστας. Αν η λίστα είναι κενή, εκτυπώνει “It is empty!” ,αλλιώς εκτυπώνει κάθε ένα στοιχείο από την αρχή μέχρι το τέλος.

Κλάση StringStackImpl

Η κλάση StringStackImpl αναπαριστά μία στοίβα από συμβολοσειρές.Για την αναπαράστασή της χρησιμοποιούμε μία μονά συνδεδεμένη λίστα.Τα πεδία της είναι η μία μονά συνδεδεμένη λίστα με το όνομα list και μία ακέραιη μεταβλητή με το όνομα size η οποία εκφράζει το μέγεθος της στοίβας.

Η μέθοδος isEmpty ελέγχει αν η στοίβα είναι άδεια εξισώνοντας το size με το 0.

Η μέθοδος push(String item) προσθέτει μία συμβολοσειρά στην κορυφή της στοίβας. Για να το κάνουμε αυτό δημιουργούμε έναν κόμβο , τον προσθέτουμε στην αρχή της λίστας list και αυξάνουμε το size κατά ένα.Η αρχή της λίστας αντιστοιχεί στην κορυφή της στοίβας και το τέλος της στη βάση της.

Η μέθοδος pop() αφαιρεί από τη στοίβα το στοιχείο το οποίο εισήχθη τελευταίο και το επιστρέφει. Για να το πετύχουμε αυτό χρησιμοποιούμε τη

μέθοδο removeFromFront() της list ,μειώνουμε το size κατά ένα και επιστρέφουμε τη συμβολοσειρά. Αν λίστα είναι άδεια ,ρίχνουμε NoSuchElementException

Η μέθοδος peek() επιστρέφει το στοιχείο το οποίο εισήχθη τελευταίο στη στοίβα χωρίς να το αφαιρέσει. Αυτό γίνεται με τη χρήση της μεθόδου returnFromFront() της list.

Η μέθοδος printStack(PrintStream stream) εκτυπώνει τα στοιχεία της στοίβας ξεκινώντας από το στοιχείο που βρίσκεται στην κορυφή. Αυτό επιτυγχάνεται με τη χρήση της μεθόδου print της λίστας list.

Η μέθοδος size() επιστρέφει το μέγεθος της στοίβας

Κλάση DoubleQueueImpl

Η κλάση DoubleQueueImpl υλοποιεί μία ουρά στην οποία αποθηκεύονται double αριθμοί. Τα πεδία της είναι μία μονά συνδεδεμένη λίστα list και μία ακέραιη μεταβλητή size.

Η μέθοδος isEmpty() ελέγχει αν η ουρά είναι άδεια.

Η μέθοδος put(Double item) προσθέτει ένα στοιχείο στο τέλος της ουράς.Αυτό συμβαίνει με τη βοήθεια της μεθόδου insertAtEnd της list .Επίσης, αυξάνουμε το size κατά ένα.

Η μέθοδος get() επιστρέφει και αφαιρεί το πρώτο στοιχείο της ουράς, δηλαδή αυτό που εισήχθη πρώτο. Γίνεται χρήση της μεθόδου removeFromFront() αφού το πιο παλιό στοιχείο της ουράς βρίσκεται στην αρχή της ουράς και το πιο καινούργιο στο τέλος της .Μειώνουμε το size κατά ένα.Αν η ουρά είναι άδεια ,η μέθοδος ρίχνει μία εξαίρεση NoSuchElementException.

Η μέθοδος peek() επιστρέφει χωρίς να αφαιρέσει το πρώτο στοιχείο της ουράς και ρίχνει εξαίρεση NoSuchElementException αν η ουρά είναι άδεια.Αυτό επιτυγχάνεται με τη χρήση της μεθόδου returnFromFront() της list.

Η μέθοδος printQueue(PrintStream stream) εκτυπώνει όλα τα στοιχεία της ουράς ξεκινώντας από αυτό που εισήχθη πρώτο.Για να συμβεί αυτό , κάνουμε χρήση της μεθόδου print της list.

Η μέθοδος size() επιστρέφει το μέγεθος της ουράς.

Θέμα Β

Η κλάση TagChecking χρησιμοποιεί ένα αντικείμενο του τύπου StringStackImpl για να ελέγξει οι ετικέτες ενός αρχείου html κλείνουν σωστά.

Το αρχείο δίνεται ως παράμετρος κατά την κλήση του προγράμματος. Για το διάβασμα του αρχείου χρησιμποποιείται ένας σαρωτής με το όνομα filescanner.Δημιουργούμε ένα αντικείμενο τύπου Pattern το οποίο αντιστοιχεί στην ετικέτα κλεισίματος και ανοίγματος στη html.Πιο συγκεκριμένα, είναι ένα μοτίβο το οποίο ξεκινάει με το '<' ,ακολουθεί μία ή καμία / ,έπειτα έχουμε ένα ή περισσότερα πεζά ή κεφαλαία λατινικά γράμματα(δε χρειάζεται να ελέγξουμε αν το αρχείο html έχει συντακτικά λάθη αλλά μόνο αν κλείνουν σωστά οι ετικέτες).Το μοτίβο κλείνει με έναν ή κανέναν αριθμό από το 1 έως το 7 και τέλος συναντάμε το σύμβολο '>'.Ελέγχουμε αν υπάρχει αυτό το μοτίβο στην τρέχουσα γραμμή. Αν η συμβολοσειρά που ταιριάζει ισούται με το '
' ή το '<img>',τότε δεν κάνουμε κάτι και απλά συνεχίζουμε .Αν η συμβολοσειρά αρχίζει με '</' ,ελέγχουμε να το στοιχείο το οποίο εισήχθη τελευταίο στη στοίβα ισούται με τη συμβολοσειρά και το αφαιρούμε αν ισχύει αυτό ,διαφορετικά ρίχνουμε μία εξαίρεση TagsNotMatchingExc. Αν δεν ισχύουν οι δύο προηγούμενες περιπτώσεις,τότε προσθέτουμε τη συμβολοσειρά στη στοίβα. Τέλος, εκτυπώνεται ΄Tags matching !' μόνο αν όλες οι ετικέτες ταιριάζουν.

Θέμα Γ

Η κλάση NetProfit υπολογίζει το καθαρό κέρδος που προκύπτει από την πώληση μετοχών. Το αρχείο στο οποίο βρίσκονται τα δεδομένα των μετοχών θα δίνεται ως παράμετρος κατά την κλάση του προγράμματος. Για την υλοποίηση αυτής της κλάσης χρησιμοποιούμε δύο αντικείμενα του τύπου DoubleQueueImpl ,το prices_queue και το amounts_queue.Στην ουρά prices_queue αποθηκεύουμε τις τιμές αγοράς των μετοχών ενώ στην amounts_queue την ποσότητα που επιθυμούμε να αγοράσουμε. Για την ανάγνωση του αρχείου χρησιμοποιούμε τον σαρωτή filescanner και δηλώνουμε μία στατική μεταβλητή η οποία συμβολίζει τη συνολική ποσότητα των μετοχών που έχουμε αγοράσει .Αρχικά, δημιουργούμε ένα loop το οποίο με τη βοήθεια του filescanner σαρώνει το αρχείο και αν η λέξη που εντοπίζει είναι η "buy",τότε προσθέτουμε τον επόμενο αριθμό στη ουρά amounts_queue και αυξάνουμε τη μεταβλητή total_amount κατά την ποσότητα που εντοπίζουμε ,ενώ αν η επόμενη λέξη ισούται με το "price",προσθέτουμε τον επόμενο αριθμό στην ουρά prices_queue.Αν η επόμενη λέξη είναι η "price", αποθηκεύουμε την ποσότητα που θέλουμε να αγοράσουμε στη μεταβλητή amount και τη τιμή πώλησης στη μεταβλητή price.Για τον υπολογισμό του κέρδους χρησιμοποιούμε τη συνάρτηση compute_profit.

Η συνάρτηση compute_profit παίρνει τέσσερα ορίσματα ,την ποσότητα που θέλουμε να πουλήσουμε ,την τιμή πώλησης, μία ουρά με double με τις ποσότητες των μετοχών και μία ουρά με double με τις τιμές των μετοχών. Αν η

στατική μεταβλητή total_amount είναι μικρότερη από την my_amount, τότε ρίχνουμε μία εξαίρεση NotEnoughStocksExc. Διαφορετικά, αφαιρούμε τα στοιχεία των δύο ουρών που εισήχθησαν τελευταία με τη μέθοδο get() και υπολογίζουμε το κέρδος. Ορίζουμε μία μεταβλητή t_amount για να αποθηκεύσουμε την τελευταία τρέχουσα ποσότητα και μία μεταβλητή t_price για να αποθηκεύσουμε την τελευταία τρέχουσα τιμή. Αν η my_amount είναι μικρότερη από την t_amount,τότε χρησιμοποιούμε την my_amount για να υπολογίσουμε το κέρδος και θέτουμε την my_amount ίση με το 0 ,αλλιώς χρησιμοποιούμε την t_amount και αφαιρούμε από την my_amount το t_amount.Τέλος, επιστρέφουμε το my_profit.