

# A

## StringStackImpl

*isEmpty()*: Επιστρέφει true αν η στοίβα είναι άδεια.

*push(String x)*: Εισάγει ένα στοιχείο τύπου string στην στοίβα.

*pop()*: Εξάγει το τελευταίο στοιχείο που μπήκε στην στοίβα, επιστρέφοντας το παράλληλα στην περίπτωση που η στοίβα είναι άδεια εμφανίζει αντίστοιχο μήνυμα.

*peek()*: Επιστρέφει το τελευταίο στοιχείο που μπήκε στην στοίβα χωρίς να το αφαιρεί . Στην περίπτωση που η στοίβα είναι αΐδια εμφανίζει αντίστοιχο μήνυμα.

*printStack()*: Εμφανίζει στο χρήστη όλα τα αντικείμενα που υπάρχουν στην στοίβα ξεκινώντας από αυτό που μπήκε τελευταίο.

*size()*: Επιστρέφει το μέγεθος της στοίβας , στην περίπτωση που είναι άδεια επιστρέφει 0.

## IntQueueImpl

*isEmpty()*: Επιστρέφει true αν η στοίβα είναι άδεια.

*put(int x)*: Εισάγει ένα στοιχείο τύπου Int στην ουρά.

*get()*: Εξάγει το πρώτο στοιχείο που μπήκε στην ουρά ενώ παράλληλα το επιστρέφει.

*peek()*: Επιστρέφει το πρώτο στοιχείο που μπήκε στην ουρά χωρίς να το αφαιρεί . Στην περίπτωση που η ουρά είναι άδεια εμφανίζει αντίστοιχο μήνυμα.

*printQueue()*: Εμφανίζει στο χρήστη όλα τα αντικείμενα που υπάρχουν στην ουρά ξεκινώντας από αυτό που μπήκε πρώτο.

*size()*: Επιστρέφει το μέγεθος της ουράς , στην περίπτωση που είναι άδεια επιστρέφει 0.

## B

### TagMatching

Αρχικά η *TagsMatching* χρησιμοποιεί την *StringStackImpl* ώστε να δημιουργήσει ένα αντικείμενο με όνομα **tagsStack**. Στην συνέχεια διαβάζει το αρχείο και ψάχνει τον πρώτο χαρακτήρα που ξεκινάει με το σύμβολο '<' όταν το βρει κοιτάζει το περιεχόμενο από το σύμβολο '<' μέχρι το σύμβολο '>' και αποθηκεύει το σύνολο αυτό '<.....>' στην στοίβα **tagsStack** στην περίπτωση όμως που μετρά το σύμβολο '<' υπάρχει το σύμβολο '/' το σύνολο '<'...>' δεν αποθηκεύεται στην στοίβα, αντ' αυτού καλείται η συνάρτηση peek() και ελέγχει αν το στοιχείο της στοίβας είναι ίδιο με το τελευταίο στοιχείο που διαβάστηκε αντικαθιστώντας το σύμβολο '/' με ". Αν τα δυο στοιχεία είναι ίδια τότε καλείται η συνάρτηση pop(). Τέλος αν η λίστα είναι άδεια σημαίνει ότι τα tags είναι matched αλλιώς δεν είναι.

## C

### NetBenefit

Το πρόγραμμα διαβάζει το αρχείο .txt και αφού είναι σε συγκεκριμένη μορφή, ελέγχει αν η γραμμή αρχίζει με την λέξη "buy" ή "sell". Στην περίπτωση που ξεκινά με την λέξη "buy", καλεί την συνάρτηση put() με όρισμα την τιμή που υπάρχει στο αρχείο (buy 50 price 25, δηλ. 25) τόσες φορές όσες είναι οι μετοχές που θέλουμε να αγοράσουμε (δηλ. 50). Στην περίπτωση που η γραμμή αρχίζει με την λέξη "sell", βρίσκει πόσες μετοχές είναι για πώληση και σε τι τιμή σε μια επαναληπτική διαδικασία, καλώντας την συνάρτηση get() αφαιρούμε την τιμή της μετοχής που είναι ήδη μέσα στην ουρά από την τιμή που θέλουμε να την πουλήσουμε. Το άθροισμα των διαφορών θα μας δώσει το τελικό κέρδος ή ζημία των αγοραπωλησιών. Η χρήση ουράς μας βοηθά αρκετά διότι δεν χρειάζεται να ενδιαφερθούμε για την σειρά που θα πουλήσουμε τις μετοχές αφού αυτές που αγοράστηκαν πρώτες θα πουληθούν πρώτες (Fist In First Out).