ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

ΕΡΓΑΣΙΑ 1Η

ΜΕΡΟΣ Α

Στο μέρος Α της εργασίας πρέπει να υλοποιηθούν οι διεπαφές της StringStack και StringQueue. Κατασκευάσαμε τις κλάσεις StringStackImpl και StringQueueImpl που υλοποιούν τις δυο παραπάνω διεπαφές αντίστοιχα. Η υλοποίηση και για τις δυο διεπαφές γίνεται με λίστα μονής σύνδεσης. Στην κλάση StringStackImpl αρχικά δηλώνουμε ένα private αντικείμενο τύπου Node με όνομα head που δείχνει στην αρχή της στοίβας δηλαδή στο τελευταίο στοιχειό που εισάγαμε στην στοίβα και που θα είναι και το πρώτο που θα αφαιρεθεί όταν καλέσουμε την μέθοδο pop. Μετά δημιουργούμε τον κατασκευαστή για τα αντικείμενα τύπου Node και στη συνεχεία δημιουργούμε και κατασκευαστή για την στοίβα μας αόριστου μεγέθους, όπου αρχικοποιούμε τον head δίνοντας του την τιμή null. Έπειτα, ορίζουμε τις μεθόδους isEmpty, push, pop, peek, printStack και size. Η μέθοδος isEmpty ελέγχει άμα η στοίβα μας είναι κενή επιστρέφοντας την τιμή true ή false. Η μέθοδος push εισάγει έναν νέο κόμβο ,που περιέχει μια μεταβλητή τύπου string ,στην στοίβα και η μέθοδος pop αφαιρεί και μας επιστρέφει τον τελευταίο κομβο που περιεχει μεταβλητή τύπου string που είχε εισαχθεί στην στοίβα μας, αρκεί η στοίβα μας να έχει στοιχειά μέσα, αλλιώς μας επιστρέφει εξαίρεση τύπου NoSuchElementException. Έπειτα, η μέθοδος peek μας επιστρέφει το πρώτο στοιχειό της στοίβας μας, δηλαδή αυτό που εισήχθη τελευταίο, χωρίς να το αφαιρεί από την στοίβα. Όπως και στην pop έτσι και εδώ άμα η στοίβα είναι κενή επιστρέφει εξαίρεση τύπου NoSuchElementException. Τέλος, έχουμε τις μεθόδους printStack όπου τυπώνονται τα στοιχειά της στοίβας άμα αυτή δεν είναι άδεια, αλλιώς τυπώνεται το κατάλληλο μήνυμα και έχουμε και την μέθοδο size όπου επιστρέφει το μέγεθος της στοίβας άμα αυτή δεν είναι άδεια, αλλιώς και εδώ τυπώνεται το κατάλληλο μήνυμα.

Στην κλάση StringQueueImpl αρχικά δηλώνουμε δύο private αντικείμενα τύπου Node με όνομα head και tail όπου το head δείχνει στην αρχή της ουράς δηλαδή στο πρώτο στοιχειό που εισάγαμε στην ουρά δηλαδή το πρώτο στοιχείο που θα αφαιρεθεί όταν καλέσουμε την μέθοδο pop. Ενώ, το tail δείχνει στο τέλος της ουράς δηλαδή εκεί που θα εισάγουμε το νέο στοιχείο της ουράς όταν καλέσουμε την μέθοδο push. Μετά δημιουργούμε

τον κατασκευαστή για τα αντικείμενα τύπου Node και στη συνεχεία δημιουργούμε και τον κατασκευαστή για την ουρά μας αόριστου μεγέθους, όπου αρχικοποιούμε τον head και τον tail δίνοντας τους την τιμή null. Έπειτα, ορίζουμε τις μεθόδους isEmpty, push, pop, peek, printQueue και size. Η μέθοδος isEmpty ελέγχει άμα η ουρά μας είναι κενή επιστρέφοντας την τιμή true ή false. Η μέθοδος push εισάγει έναν νέο κόμβο ,που περιέχει μια μεταβλητή τύπου string ,στο τέλος της ουράς μας, δηλαδή μετά τον κόμβο στον οποίο δείχνει το tail. Έτσι όταν εισαχτεί ο νέος κόμβος το tail θα δείχνει σε αυτόν. Η μέθοδος pop αφαιρεί και μας επιστρέφει τον πρώτο κόμβο, που περιέχει μεταβλητή τύπου string, που είχε εισαχθεί στην ουρά μας, αρκεί η ουρά μας να έχει στοιχειά μέσα, αλλιώς μας επιστρέφει εξαίρεση τύπου NoSuchElementException. Έπειτα, η μέθοδος peek μας επιστρέφει το πρώτο στοιχειό που εισήχθη στην ουρά μας χωρίς να το αφαιρεί από την ουρά. Όπως και στην pop έτσι και εδώ άμα η ουρά είναι κενή επιστρέφει εξαίρεση τύπου NoSuchElementException. Τέλος, έχουμε τις μεθόδους printQueue όπου τυπώνονται τα στοιχειά της ουράς άμα αυτή δεν είναι άδεια, αλλιώς τυπώνεται το κατάλληλο μήνυμα και έχουμε και την μέθοδο size όπου επιστρέφει το μέγεθος της ουράς άμα αυτή δεν είναι άδεια, αλλιώς και εδώ τυπώνεται το κατάλληλο μήνυμα.

ΜΕΡΟΣ Β

Στο Β μέρος της εργασίας πρέπει να διαβάζουμε από ένα αρχείο τα στοιχεία του και να τα αποθηκεύουμε σε έναν πίνακα χαρακτήρων. Ο πίνακας αναπαριστά έναν λαβύρινθο ο οποίος απαρτίζεται μόνο από μηδέν(0) και ένα (1) όπου όπου έχει ένα υπάρχει εμπόδιο και όπου έχει μηδέν είναι μονοπάτι όπου μπορεί να περπατηθεί. Αφού διαβάσουμε το αρχείο, δημιουργήσουμε τον πίνακα και κάνουμε κάποιους ελέγχους για το αν έχουμε είσοδο στον λαβύρινθο, αν έχουμε παραπάνω από μια είσοδο άμα όλα τα δεδομένα του πίνακα είναι σωστά, άμα έχουμε τον σωστό αριθμό γραμμών και στήλεων τότε αρχίζουμε και αναζητάμε την έξοδο από τον λαβύρινθο. Αρχικά, αποθηκεύουμε στην στοίβα μας με όνομα StackTrace τις συντεταγμένες εισόδου στον λαβύρινθο καλώντας την μέθοδο push της κλάσης StringStackImpl και μετά κινούμαστε είτε δεξιά είτε αριστερά είτε πάνω είτε κάτω, δηλαδή είτε στην προηγούμενη γραμμή είτε στην επόμενη είτε στην προηγούμενη στήλη είτε στην επόμενη. Προϋπόθεση για να επιλέξουμε που θα κινηθούμε είναι ώστε το στοιχειό του πίνακα στην θέση που θέλουμε να πάμε να ισούται με μηδέν. Άμα, κινηθούμε τότε οι συντεταγμένες του νέου κελιού του πίνακα που κινηθήκαμε αποθηκεύονται στην στοίβα καλώντας την μέθοδο push της StringStackImpl και το στοιχείο

στο κελί που κινηθήκαμε παίρνει την τιμή X ώστε να ξέρουμε ότι έχουμε ήδη περάσει από αυτό το κελί. Στην περίπτωση που έχουμε κινηθεί τόσο ώστε να έχουμε φτάσει είτε στην πρώτη γραμμή είτε στην τελευταία γραμμή είτε στην πρώτη στήλη είτε στην τελευταία στήλη αυτό σημαίνει ότι βρήκαμε μια έξοδο από τον λαβύρινθο και το πρόγραμμα μας τελειώνει καλώντας την μέθοδο printStack για να εμφανίσει τις συντεταγμένες όλων των κελιών από τα οποία περάσαμε για να φτάσουμε μέχρι την έξοδο. Στην περίπτωση που έχουμε κινηθεί και ξαφνικά βρεθήκαμε σε αδιέξοδο, δηλαδή, οι επιλογές μας είναι μόνο κελιά με τιμή 1 είτε με X που σημαίνει ότι έχουμε περάσει ήδη, τότε αρχίζουμε και κινούμαστε προς τα πίσω, ώστε να βρούμε ένα νέο μονοπάτι για να κινηθούμε, με τον εξής τρόπο. Καλούμε την μέθοδο pop της StringStackImpl ώστε να αφαιρέσουμε από την στοίβα τις συντεταγμένες του κελιού που είμαστε και να πάμε στο προηγούμενο κελί. Έπειτα, καλώντας την μέθοδο peek της StringStackImpl παίρνουμε τις συντεταγμένες του προηγούμενου κελιού που ήμασταν και ελέγχουμε άμα υπάρχει άλλο μονοπάτι. Άμα πάλι δεν υπάρχει άλλο κελί στο οποίο μπορούμε να κινηθούμε συνεχίζουμε την ίδια διαδικασία μέχρι να βρούμε ένα νέο κελί. Προϋπόθεση για να κάνουμε όλα αυτά τα βήματα είναι να μην είναι άδεια η στοίβα μας. Άμα τελικά έχουμε φτάσει σε αδιέξοδο και έχουμε επιστρέψει και στα προηγούμενα κελιά που ήμασταν και δεν έχουμε βρει ένα νέο μονοπάτι ή ένα μονοπάτι εξόδου από το λαβύρινθο τότε το πρόγραμμα μας τερματίζει και εμφανίζει και το κατάλληλο μήνυμα ότι δεν υπάρχει έξοδος. Αυτό σημαίνει ότι έχουμε κινηθεί τόσο πίσω ώστε να έχουμε φτάσει στην είσοδο του λαβύρινθου μας όπου έχουμε ψάξει και εκεί για ένα νέο κελί στο οποίο θα μπορούσαμε να κινηθούμε αλλά δεν υπάρχει κανένα οπότε έχουμε καλέσει την μέθοδο pop της StringStackImpl και οπότε έχει αδειάσει όλη η στοίβα μας.

ΜΕΡΟΣ Γ

Στην κλάση StringQueueImpl υλοποιούμε την διεπαφή StringQueue του μέρους Α αλλά αυτή τη φορά χρησιμοποιούμε λίστα κυκλικής σύνδεσης. Αρχικά δηλώνουμε ένα private αντικείμενο τύπου Node με όνομα tail το οποίο δείχνει στο τέλος της ουράς δηλαδή εκεί που θα εισάγουμε το νέο στοιχείο της ουράς όταν καλέσουμε την μέθοδο push. Επειδή όμως η ουρά μας υλοποιείται με λίστα κυκλικής σύνδεσης αυτό σημαίνει ότι το next του tail δείχνει στο αρχικό στοιχείο που εισήχθη στην ουρά Μετά δημιουργούμε τον κατασκευαστή για τα αντικείμενα τύπου Node και στη συνεχεία δημιουργούμε και τον κατασκευαστή για την ουρά μας αόριστου μεγέθους, όπου αρχικοποιούμε τον tail δίνοντας του την τιμή null. Έπειτα, ορίζουμε

τις μεθόδους isEmpty, push, pop, peek, printQueue και size. Η μέθοδος isEmpty ελέγχει άμα η ουρά μας είναι κενή επιστρέφοντας την τιμή true ή false. Η μέθοδος push εισάγει έναν νέο κόμβο ,που περιέχει μια μεταβλητή τύπου string ,στο τέλος της ουράς μας, δηλαδή μετά τον κόμβο στον οποίο δείχνει το tail. Έτσι όταν εισαχτεί ο νέος κόμβος το tail θα δείχνει σε αυτόν και το next του νέου κόμβου στον πρώτο που είχε εισαχθεί δηλαδή εκεί που έδειχνε παλιά το next του tail. Η μέθοδος pop αφαιρεί και μας επιστρέφει τον πρώτο κόμβο, που περιέχει μεταβλητή τύπου string, που είχε εισαχθεί στην ουρά μας, αρκεί η ουρά μας να έχει στοιχειά μέσα, αλλιώς μας επιστρέφει εξαίρεση τύπου NoSuchElementException. Δηλαδή, αφαιρείται ο κόμβος στον οποίο δείχνει το next.tail . Έπειτα, η μέθοδος peek μας επιστρέφει το πρώτο στοιχειό που εισήχθη στην ουρά μας χωρίς να το αφαιρεί από την ουρά. Όπως και στην pop έτσι και εδώ άμα η ουρά είναι κενή επιστρέφει εξαίρεση τύπου NoSuchElementException. Τέλος, έχουμε τις μεθόδους printQueue όπου τυπώνονται τα στοιχειά της ουράς άμα αυτή δεν είναι άδεια, αλλιώς τυπώνεται το κατάλληλο μήνυμα και έχουμε και την μέθοδο size όπου επιστρέφει το μέγεθος της ουράς άμα αυτή δεν είναι άδεια, αλλιώς και εδώ τυπώνεται το κατάλληλο μήνυμα.