

### Οδηγίες Υλοποίησης:

#### Μέρος Α

Χρησιμοποιήσαμε μια συνδεδεμένη λίστα “LinkedList” , η οποία υλοποιήθηκε με Generics όπως και η “StringStack” και η “StringQueue” , για την αποφυγή χρήσης αντικειμένων τύπου Object και την ανεξαρτησία αυτών των δομών από τον τύπο των δεδομένων τους.

LinkedList:Η υλοποίησή της έγινε με την χρήση δύο αναφορών “head”, “tail” οι οποίοι δείχνουν ο ένας στην αρχή της λίστας και ο άλλος στο τέλος της αντίστοιχα. Η αναφορά “tail” μας βοηθάει να προσθέτουμε στοιχεία στο τέλος της λίστας σε χρόνο  $O(1)!$

Οι StringStack , StringQueue έχουν υλοποιηθεί με την LinkedList έτσι ώστε όλες οι μέθοδοι τους για την πρόσθεση και την αφαίρεση στοιχείων ( push,pop,put,get,peek ) σε σταθερό χρόνο  $O(1)!$

#### Μέρος Β

Η υλοποίησή της κλάσης του Thissea αποτελείται από δύο μέρη. Ένα μέρος χρησιμεύει στο διάβασμα του αρχείου που περιέχει τον λαβύρινθο και το άλλο για την επίλυση του. Η μέθοδος “loadFile” αρχικά τοποθετεί σε έναν πίνακα χαρακτήρων τον λαβύρινθο του αρχείου λαμβάνοντας υπόψη κάθε δυνατή περίπτωση σφάλματος και εμφανίζοντας και το κατάλληλο μήνυμα για εκσφαλμάτωση. Η μέθοδος “ findPath ” είναι αναδρομική και δέχεται ως όρισμα το σημείο εκκίνησης και εξετάζει αναδρομικά όλα τα πιθανά μονοπάτια που ως αρχή το σημείο αυτό. Κάθε φορά που κατευθυνόμαστε σε σημείο το οποίο είναι εντός ορίων του πίνακα και είναι μηδέν κάνουμε τις συντεταγμένες του “push” στο “Stack” και το μετατρέπουμε από μηδέν σε ένα για να αποφύγουμε να διασχίσουμε ξανά το σημείο αυτό . Αν ένα μονοπάτι διαπιστωθεί ότι οδηγεί σε αδιέξοδο, δηλαδή δεν μπορούμε να κινηθούμε προς καμία άλλη κατεύθυνση τότε κάνουμε “pop” από τη “Stack” το σημείο αυτό.Καθώς τερματίζονται αναδρομικά τα Stack Frames γίνονται “pop” όλα τα άλλα σημεία του μονοπατιού αυτού μέχρις ότου καταλήξουμε σε ένα σημείο από το οποίο μπορούμε να ξεκινήσουμε ένα καινούργιο μονοπάτι (backtracking). Η μέθοδος τερματίζεται όταν βρεθεί η πρώτη έξοδος γίνουν “push” οι συντεταγμένες της στη στοίβα και επιστρέφει «true»,διαφορετικά όταν διαπιστωθεί ότι δεν υπάρχει καμία έξοδος επιστρέφει «false».

Η μέθοδος “solveMaze” ανάλογα με την τιμή που της επιστράφηκε εμφανίζει είτε τις συντεταγμένες της εξόδου είτε κατάλληλο μήνυμα.

#### Μέρος Γ

Σε αυτό το μέρος χρησιμοποιήσαμε μια κυκλική λίστα “CircularList”.Για την υλοποίησή της χρησιμοποιήθηκε μια αναφορά “tail” η οποία δείχνει στο αντικείμενο που προστέθηκε τελευταίο στην λίστα έτσι ώστε όταν προσθέτουμε ένα αντικείμενο η ουρά να δείχνει στο νέο αυτό αντικείμενο “temp”. Η αναφορά “next” του “temp” με κατάλληλες αναθέσεις δείχνει στο αντικείμενο το οποίο προστέθηκε πρώτο στη λίστα έτσι ώστε να πετυχαίνουμε πρόσθεση και αφαίρεση (insert, remove) σε σταθερό χρόνο. Κατά συνέπεια για την υλοποίηση της ουράς του Γ ερωτήματος “StringQueueWithOnePointer” χρησιμοποιήθηκε η “CircularList” .