Δομές Δεδομένων
Χειμερινό εξάμηνο 2022
Εργασία 1
Ουρές: Υλοποιήσεις ΑΤΔ και εφαρμογές

Ονοματεπώνυμο:	Αριθμός Μητρώου:
Γεώργιος Βεάζογλου	3190347
Βασιλική Δρόσου	3200044

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

1. *Πως υλοποιήσαμε τις διεπαφές για τη στοίβα στο μέρος Α:*

- *isEmpty():* Ελέγχει αν το πρώτο κελί της στοίβας είναι κενό. Αν ναι, τότε δεν υπάρχει κανένα στοιχείο στη στοίβα.
- *push():* Προσθέτει ένα στοιχείο στην αρχή της στοίβας και στη συνέχεια αυξάνει τον μετρητή size (που ήταν αρχικοποιημένος με μηδέν) κατά 1.
- *pop():* Αρχικά ελέγχει μέσω της μεθόδου *isEmpty()* αν η στοίβα είναι κενή. Αν δεν είναι, αφαιρεί τον κόμβο από την αρχή της στοίβας, επιστρέφει τα στοιχεία του κόμβου και μειώνει τον μετρητή size κατά 1. Αν η στοίβα είναι κενή, εμφανίζει error (η μέθοδος πετάει εξαίρεση τύπου NoSuchElementException).
- *peek():* Αρχικά ελέγχει μέσω της μεθόδου *isEmpty()* αν η στοίβα είναι κενή. Αν δεν είναι, επιστέφει τα δεδομένα του πρώτου κόμβου. Αν η στοίβα είναι κενή, εμφανίζει error (η μέθοδος πετάει εξαίρεση τύπου NoSuchElementException).
- *printStack():* Αρχικά ελέγχει μέσω της μεθόδου *isEmpty()* αν η στοίβα είναι κενή. Αν δεν είναι, την εμφανίζει στην οθόνη. Διαφορετικά, ενημερώνει τον χρήστη ότι η στοίβα είναι άδεια, εμφανίζοντας κατάλληλο μνήμα στον χρήστη.
- *size():* Επιστρέφει την τιμή του μετρητή size, δηλαδή το μέγεθος της στοίβας.

Πως υλοποιήσαμε τις διεπαφές για την ουρά στο μέρος Α:

- *isEmpty():* Ελέγχει αν το πρώτο κελί της ουράς είναι κενό. Αν ναι, τότε δεν υπάρχει κανένα στοιχείο στην ουρά.
- *put():* Εισάγει έναν κόμβο στο τέλος της ουράς και στη συνέχεια αυξάνει τον μετρητή size (που ήταν αρχικοποιημένος με μηδέν) κατά 1.
- *get():* Αρχικά ελέγχει μέσω της μεθόδου *isEmpty()* αν η ουρά είναι κενή. Αν δεν είναι, αφαιρεί τον κόμβο από την αρχή της ουράς, επιστρέφει τα στοιχεία του κόμβου και μειώνει τον μετρητή size κατά 1. Αν η ουρά είναι κενή, εμφανίζει error (η μέθοδος πετάει εξαίρεση τύπου NoSuchElementException).
- *peek():* Αρχικά ελέγχει μέσω της μεθόδου *isEmpty()* αν η ουρά είναι κενή. Αν δεν είναι, επιστέφει τα δεδομένα του πρώτου κόμβου. Αν η ουρά είναι κενή, εμφανίζει error (η μέθοδος πετάει εξαίρεση τύπου NoSuchElementException).
- *printQueue():* Αρχικά ελέγχει μέσω της μεθόδου *isEmpty()* αν η ουρά είναι κενή. Αν δεν είναι, την εμφανίζει στην οθόνη. Διαφορετικά, ενημερώνει τον χρήστη ότι η ουρά είναι άδεια, εμφανίζοντας κατάλληλο μνήμα στον χρήστη.
- *size():* Επιστρέφει την τιμή του μετρητή size, δηλαδή το μέγεθος της ουράς.

2. *Πως χρησιμοποιήσαμε την υλοποίηση από το μέρος Α για να φτιάξουμε το πρόγραμμα που ζητείται:*

Κατά την εκκίνηση του προγράμματος, καλείται η μέθοδος createMaze, η οποία παίρνει σαν όρισμα το αρχείο που υποδεικνύουμε κατά την εκτέλεση. Έπειτα, η μέθοδος διαβάζει τις γραμμές του txt αρχείου και αποθηκεύει τις διαστάσεις του λαβυρίνθου (πρώτη γραμμή) και τις συντεταγμένες της εισόδου (δεύτερη γραμμή). Επίσης περνάει τα στοιχεία του λαβυρίνθου στον δισδιάστατο πίνακα maze που αποτελείται από String στοιχεία. Γίνεται έλεγχος για τις διαστάσεις και τις συντεταγμένες που δόθηκαν και, εφόσον είναι σωστές, καλείται η μέθοδος findExit. Έτσι γίνονται push στη στοίβα οι συντεταγμένες της εισόδου Ε και μπαίνουμε σε ένα while loop με τη συνθήκη ότι η στοίβα δεν είναι άδεια. Στη συνέχεια κάνουμε pop και αποθηκεύουμε τις συντεταγμένες σε ένα νέο πίνακα που διευκολύνει στην κλήση μεθόδων. Ελέγχουμε αν οι συντεταγμένες ικανοποιούν τις απαιτήσεις για να αποτελέσουν έξοδο με τη μέθοδο mazeSolved. Εφόσον δεν αποτελούν έξοδο, αλλάζουμε το περιεχόμενο σε V για να το σημειώσουμε ως εξετασμένο. Ελέγχουμε σε ποια κατεύθυνση μπορεί να υπάρξει έξοδος με τη χρήση της nextPath. Αν βρεθεί έξοδος, ο χρήστης ενημερώνεται αναλόγως, αλλιώς η στοίβα αδειάζει και τερματίζει το πρόγραμμα (δεν βρέθηκε έξοδος).