ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
SCHOOL OF INFORMATION SCIENCES & TECHNOLOGY
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
DEPARTMENT OF INFORMATICS

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Διδάσκουσα: Κ. Παπακωνσταντινοπούλου

Χειμερινό εξάμηνο 2022

Εργασία 1
Ουρές: Υλοποιήσεις ΑΤΔ και εφαρμογές
Προθεσμία υποβολής: 27/11/2022, 23:59

Σκοπός της εργασίας είναι η εξοικείωση με βασικούς αφηρημένους τύπους δεδομένων όπως οι στοίβες, και οι ουρές FIFO. Η εργασία αποτελείται από 2 υλοποιήσεις ΑΤΔ (Μερη Α και Γ) καθώς και 1 εφαρμογή (Μέρος Β). Διαβάστε προσεκτικά την εκφώνηση και τα ζητούμενα της εργασίας.

Μέρος Α (30 μονάδες)

Στο φάκελο «Εγγραφα/Εργασίες» του eclass, δίνονται οι διεπαφές StringStack και StringQueue, που δηλώνουν τις βασικές μεθόδους για μια στοίβα και μια ουρά FIFO, με στοιχεία τύπου String. Δημιουργήστε μία υλοποίηση των ΑΤΔ StringStack και StringQueue, δηλαδή γράψτε 2 κλάσεις που υλοποιούν τις 2 διεπαφές.

Οδηγίες υλοποίησης:

- Οι κλάσεις σας πρέπει να λέγονται StringStackImpl και StringQueueImpl.
- Η υλοποίηση και για τις 2 διεπαφές θα πρέπει να γίνει χρησιμοποιώντας λίστα μονής σύνδεσης.
- Κάθε πράξη εισαγωγής ή εξαγωγής στοιχείου (δηλαδή κάθε εκτέλεση των μεθόδων push και pop στη στοίβα, και put και get στην ουρά FIFO) θα πρέπει να ολοκληρώνεται σε χρόνο $O(1)$, δηλαδή σε χρόνο ανεξάρτητο από τον αριθμό των αντικειμένων που είναι μέσα στην ουρά. Ομοίως, η μέθοδος size θα πρέπει να εκτελείται σε $O(1)$.
- Όταν η στοίβα ή η ουρά είναι άδεια, οι μέθοδοι που διαβάζουν από την δομή θα πρέπει να πετάνε εξαίρεση τύπου NoSuchElementException. Η εξαίρεση NoSuchElementException ανήκει στην core βιβλιοθήκη της Java. Κάντε την import από το πακέτο java.util. Μην κατασκευάσετε δική σας εξαίρεση.
- Μπορείτε να χρησιμοποιήσετε τη λίστα μονής σύνδεσης που παρουσιάστηκε στο φροντιστήριο του μαθήματος ή να γράψετε εξ' ολοκλήρου τη δική σας λίστα ή να χρησιμοποιήσετε μόνο αντικείμενα τύπου Node μέσα στην κλάση της στοίβας/ουράς. Για να αποκτήσετε καλύτερη εξοικείωση προτείνουμε να ξεκινήσετε από την αρχή και να γράψετε τις δικές σας κλάσεις (σίγουρα δεν θα χάσετε μονάδες όμως χρησιμοποιώντας ό,τι έχετε δει στο εργαστήριο).
- Προαιρετικά: μπορείτε να κάνετε την υλοποίηση σας με χρήση generics για να μπορείτε να χειρίζεστε στοίβες και ουρές με οποιοδηποτε τύπο αντικειμένων. Υπάρχει ένα 10% βονυς σε όσους χρησιμοποιήσουν generics για την εργασία.
- Δεν επιτρέπεται να χρησιμοποιήσετε έτοιμες υλοποιήσεις δομών τύπου λίστας, στοίβας, ουράς, από την βιβλιοθήκη της Java (π.χ. Vector, ArrayList κλπ).

Μέρος Β (30 μονάδες)

Χρησιμοποιώντας την υλοποίηση της στοίβας του μέρους Α, γράψτε ένα πρόγραμμα-πελάτη το οποίο θα κάνει διάσχιση σε έναν λαβύρινθο με σκοπό να βρίσκει την έξοδο. Το πρόγραμμά σας θα πρέπει να παίρνει ως είσοδο ένα .txt αρχείο, το οποίο θα περιέχει τον λαβύρινθο σε μορφή πίνακα χαρακτήρων, διαστάσεων $n \times m$, όπου n, m, ακέραιοι. Ο πίνακας μπορεί να περιέχει μόνο τους χαρακτηρες 0, 1 και E, όπου το E θα βρίσκεται μόνο σε ένα σημείο του πίνακα και θα υποδηλώνει την είσοδο στο λαβύρινθο. Όταν κάποιος μπαίνει στον λαβύρινθο, μπορεί να κινηθεί οριζόντια ή κάθετα (αλλά όχι διαγώνια) προς οποιαδήποτε κατεύθυνση που περιέχει 0 (δείτε το παράδειγμα παρακάτω). Αν φτάσετε στα σύνορα του πίνακα (πρώτη ή τελευταία γραμμή και πρώτη ή τελευταία στήλη), και βρείτε 0, τότε έχετε φτάσει σε μια έξοδο του λαβυρίνθου. Είναι δυνατόν να υπάρχουν πολλαπλές έξοδοι στο λαβύρινθο (ή και καμία). Το πρόγραμμά σας θα πρέπει να τυπώνει τις συντεταγμένες της εξόδου που βρήκε, ενώ αν δεν υπάρχει τρόπος διαφυγής, θα πρέπει να τυπώνει αντίστοιχο μήνυμα.

Παράδειγμα: Η είσοδός θα είναι ένα αρχείο στην μορφή του παρακάτω παραδείγματος

```
9 7
0 3
1 1 1 E 1 1 1
1 1 1 0 1 1 1
1 0 0 0 1 0 1
1 0 1 0 1 0 0
1 1 1 0 1 1 1
1 0 0 0 0 0 1
1 0 1 1 1 0 1
1 0 1 1 0 0 1
0 1 1 1 0 1 1
```

Στην πρώτη γραμμή δηλώνονται οι διαστάσεις του λαβυρίνθου (εδώ $n = 9$, $m = 7$). Στην δεύτερη γραμμή ακολουθούν οι συντεταγμένες του σημείου εισόδου, όπου εδώ είναι στην γραμμή 0 και στην στήλη 3 (υποθέτουμε ότι ονομάζουμε τις γραμμές από 0 ως $n - 1$ και τις στήλες από 0 ως $m - 1$). Στο παράδειγμα αυτό, η εξερεύνηση θα ξεκινήσει κινούμενη προς τα κάτω. Αν στρίψετε αριστερά (όπως κοιτάμε τον πίνακα), θα δείτε ότι θα φτάσετε σύντομα σε αδιέξοδο και θα πρέπει να γυρίσετε πίσω. Εν τέλει, συνεχίζοντας το ψάξιμο, μια έξοδος που μπορείτε να φτάσετε είναι στις συντεταγμένες (8, 4).

Οδηγίες υλοποίησης:

- Το πρόγραμμα σας πρέπει να λέγεται Thiseas.java.
- Θα πρέπει να κάνετε χρήση της υλοποίησης της στοίβας από το Μέρος Α. Η χρήση της στοίβας ενδείκνυται για να μπορέσετε να υλοποιήσετε την αναζήτηση της εξόδου με backtracking. Σκεφτείτε τι πρέπει να κάνετε όταν φτάνετε σε αδιέξοδο, και με ποιο τρόπο θα μπορέσετε να συνεχίσετε την αναζήτηση.
- Για να ελέγξετε την ορθότητα του προγράμματος σας, είναι καλό να φτιάξετε μερικούς διαφορετικούς λαβυρίνθους με διαφορετικά χαρακτηριστικά (π.χ. με πολλαπλές εξόδους, με καμία έξοδο, με πιο μεγάλες διαστάσεις, κτλ) και να τρέξετε τον κώδικά σας με αυτές τις εισόδους.
- Είναι επιτρεπτό, αν αποθηκεύσετε τον λαβύρινθο σε πίνακα χαρακτήρων, να κάνετε μετέπειτα αλλαγές πάνω στα στοιχεία του πίνακα (π.χ. αν θέλετε να χρησιμοποιήσετε κάποιον άλλο χαρακτήρα για να δείξετε ότι έχετε ήδη επισκεφτεί κάποια θέση κατά τη διάρκεια εκτέλεσης του προγράμματος). Πέρα από αυτό όμως, θα πρέπει να χρησιμοποιήσετε κατάλληλα τη στοίβα του Μέρους Α.
- Η εργασία σας θα εξεταστεί σε εισόδους της παραπάνω μορφής. Αν υπάρχει κάποιο λάθος στα δεδομένα εισόδου, το πρόγραμμα πρέπει να τερματίζει τυπώνοντας αντίστοιχο μήνυμα (π.χ. αν οι γραμμές και στήλες που διαβάσατε στην πρώτη σειρά εισόδου, δεν ταιριάζουν με τον πίνακα που διαβάζετε μετά ή αν δεν υπάρχει E στον λαβύρινθο, κτλ).
- Πρέπει να δίνετε ως όρισμα ολο το μονοπάτι για το .txt αρχείο εισόδου, π.χ. αν το τρέξετε από τη γραμμή εντολών, η εκτέλεση του προγράμματος θα είναι ως εξής:
  $>$ java Thiseas path_to_file/filename.txt

Μέρος Γ (30 μονάδες)

Η υλοποίηση της διεπαφής StringQueue με λίστα μονής σύνδεσης στο μέρος Α, θα πρέπει αναγκαστικά να χρησιμοποιεί 2 μεταβλητές σαν δείκτες στην κεφαλή και στο τέλος, τους head και tail, για να μπορείτε να εκτελείτε σωστά τις εισαγωγές και εξαγωγές στοιχείων, όπως είδαμε και στο μάθημα. Το ζητούμενο στο μέρος Γ είναι να φτιάξετε μία νέα υλοποίηση της ουράς FIFO, χρησιμοποιώντας μόνο έναν από τους δείκτες αυτούς.
Υπόδειξη: Χρησιμοποιήστε κυκλική λίστα αντί για λίστα μονής σύνδεσης.

Οδηγίες υλοποίησης:

- Η κλάση σας πρέπει να λέγεται StringQueueWithOnePointer.java.
- Οι λειτουργίες εισαγωγής και εξαγωγής θα πρέπει να γίνονται σε $O(1)$ και γενικότερα ισχύουν και εδώ όλες οι οδηγίες που αναγράφονται για το Μέρος Α.

Μέρος Δ - Αναφορά παράδοσης (10 μονάδες)

Ετοιμάστε μία σύντομη αναφορά σε pdf αρχείο (μην παραδώσετε Word ή txt αρχεία!) με όνομα project1-report.pdf, στην οποία θα αναφερθείτε στα εξής:

1. Εξηγήστε συνοπτικά πώς υλοποιήσατε τις διεπαφές στα μέρη Α και Γ. Ειδικά για το Μέρος Γ, εξηγήστε την ιδέα με την οποία μπορείτε να αποφύγετε τη χρήση 2 δεικτών για την αρχή και το τέλος της ουράς (άνω όριο 3 σελίδες).
2. Για το μέρος Β, εξηγήστε πώς χρησιμοποιήσατε την υλοποίηση από το μέρος Α για να φτιάξετε το πρόγραμμα που ζητείται (άνω όριο 3 σελίδες).

Οδηγίες Παράδοσης

Η εργασία σας θα πρέπει να μην έχει συντακτικά λάθη και να μπορεί να μεταγλωττίζεται. Εργασίες που δεν μεταγλωττίζονται χάνουν το 50% της συνολικής αξίας.

Η εργασία θα αποτελείται από:

1. Τον πηγαίο κώδικα (source code). Τοποθετήστε σε ένα φάκελο με όνομα src τα αρχεία java που έχετε φτιάξει. Χρησιμοποιήστε τα ονόματα των κλάσεων όπως ακριβώς δίνονται στην εκφώνηση. Επιπλέον, φροντίστε να συμπεριλάβετε όποια άλλα αρχεία πηγαίου κώδικα απαιτούνται για να μεταγλωττίζεται η εργασία σας. Φροντίστε επίσης να προσθέσετε επεξηγηματικά σχόλια όπου κρίνετε απαραίτητο στον κώδικά σας.
2. Την αναφορά παράδοσης.

΄Ολα τα παραπάνω αρχεία θα πρέπει να μπουν σε ένα αρχείο zip. Το όνομα που θα δώσετε στο αρχείο αυτό θα είναι ο αριθμός μητρώου σας πχ. 3030056_3030066.zip ή 3030056.zip (αν δεν είστε σε ομάδα). Στη συνέχεια, θα υποβάλετε το zip αρχείο σας στην περιοχή του μαθήματος «Εργασίες» στο e-class. Δεν χρειάζεται υποβολή και από τους 2 συμμετέχοντες μιας ομάδας.