

**Πανεπιστήμιο Ιωαννίνων – Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Δομές Δεδομένων [ΜΥΥ303] – Χειμερινό Εξάμηνο 2020**

**3^η Εργαστηριακή Άσκηση
Στοιίβες και Ουρές**

Παράδοση έως Πέμπτη 12/11, 12:00 από το eCourse

ΠΡΟΣΟΧΗ: Γράψτε σε κάθε αρχείο που παραδίδετε τα ονόματα και τους Α.Μ. των μελών της ομάδας σας. Συμπεριλάβετε όλα τα αρχεία σας (κώδικας Java και lab3results.txt) σε ένα zip αρχείο το οποίο να έχει το όνομα ενός μέλους της ομάδας σας με λατινικούς χαρακτήρες.

Μας δίνεται ένα ημιτελές πρόγραμμα Parentheses.java το οποίο πρέπει να συμπληρώσουμε ώστε να επιτελεί την ακόλουθη λειτουργία. Το πρόγραμμα δέχεται στην είσοδο μια ακολουθία που αποτελείται από παρενθέσεις τριών τύπων: `()`, `[]` και `{}`. Σκοπός του είναι να ελέγξει αν η ακολουθία έχει ορθή σύνταξη, δηλαδή αν οι παρενθέσεις είναι φωλιασμένες σωστά. Επιπλέον, πρέπει να αποθηκεύει τις θέσεις των αντίστοιχων αριστερών και δεξιών παρενθέσεων, εφόσον η ακολουθία είναι συντακτικά ορθή.

Για παράδειγμα, η ακολουθία `[({ }) { [] () }] { [()] }` είναι έγκυρη, όπως φαίνεται από το χρωματισμό των αντίστοιχων παρενθέσεων `[({ }) { [] () }] { [()] }`, οπότε έχουμε τις ακόλουθες αντιστοιχίσεις: 2-3 `{}`, 1-4 `()`, 6-7 `[]`, 8-9 `()`, 5-10 `{}`, 0-11 `[]`, 14-15 `()`, 13-16 `[]` και 12-17 `{}`. Δηλαδή η παρένθεση `{` στη θέση 2 της ακολουθίας αντιστοιχεί στην παρένθεση `}` στη θέση 3 της ακολουθίας κοκ. (Προσέξτε ότι η πρώτη παρένθεση βρίσκεται στη θέση 0.) Αντίθετα, ακολουθίες όπως `{ () } ,) (, [()` και `[] ()` δεν είναι έγκυρες.

Ο παραπάνω έλεγχος μπορεί να γίνει με τη βοήθεια μιας στοιίβας, η οποία θα αποθηκεύει τις αριστερές παρενθέσεις μαζί με τον αριθμό της θέσης που βρίσκονται στην ακολουθία. Για το σκοπό αυτό, θα χρησιμοποιήσουμε την κλάση `Pair.java`, η οποία δημιουργεί ζεύγη `<c, i>` που αποτελούνται από ένα χαρακτήρα `c` και έναν ακέραιο `i`. Η αποθήκευση των θέσεων των αντίστοιχων αριστερών και δεξιών παρενθέσεων θα γίνει σε μία FIFO ουρά ακέραιων αριθμών.

Συγκεκριμένα, πρέπει να υλοποιήσετε (στο αρχείο `Parentheses.java`) τη μέθοδο `void ParseInput(String str)` η οποία εκτελεί την ακόλουθη λειτουργία. Η μέθοδος διαβάζει διαδοχικά τους χαρακτήρες της ακολουθίας εισόδου `str`, από τα αριστερά (θέση 0) προς τα δεξιά. Έστω ότι βρισκόμαστε στη θέση `i` όπου διαβάζουμε τον χαρακτήρα `c`. Τότε κάνουμε ένα από τα παρακάτω:

- Αν ο `c` δεν είναι παρένθεση τότε τον αγνοούμε.
- Αν ο `c` είναι αριστερή παρένθεση, `(`, `[` ή `{`, τότε ωθούμε στη στοιίβα το ζεύγος `<c, i>`.
- Αν ο `c` είναι δεξιά παρένθεση, `)`, `]`, ή `}`, τότε απωθούμε από τη στοιίβα το ζεύγος `<d, j>` που βρίσκεται στην κορυφή της. Ο χαρακτήρας `d` είναι αριστερή παρένθεση (αφού μόνο αυτοί οι χαρακτήρες ωθούνται στη στοιίβα), οπότε ελέγχουμε αν είναι του ίδιου τύπου με τον `c`, π.χ., `d = {` και `c = }`. Αν είναι, τότε τοποθετούμε στο τέλος της FIFO ουράς τους αριθμούς `j` και `i`, οι οποίοι αντιστοιχούν στο διάστημα που καλύπτουν οι παρενθέσεις `d` και `c`. Διαφορετικά δηλώνουμε ότι υπάρχει συντακτικό λάθος και διακόπτουμε την εκτέλεση του προγράμματος.

Ζητούμενες Μέθοδοι

- Υλοποιήστε στα αρχεία `Stack.java` και `Queue.java` μια στοιίβα και μια FIFO ουρά, αντίστοιχα, γενικού τύπου με τη χρήση Java generics. Μπορείτε να

Πανεπιστήμιο Ιωαννίνων – Τμήμα Μηχανικών Η/Υ και Πληροφορικής Δομές Δεδομένων [ΜΥΥ303] – Χειμερινό Εξάμηνο 2020

χρησιμοποιήσετε υλοποιήσεις της αρεσκείας σας, είτε με πίνακα είτε με συνδεδεμένη λίστα.

- Συμπληρώστε στο αρχείο `Parentheses.java` τις μεθόδους `void ParseInput(String str)` και `void PrintOutput(Queue<Int> Q)`. Η `ParseInput` λαμβάνει ως όρισμα την ακολουθία εισόδου `str`, κάνει το συντακτικό έλεγχο με τη βοήθεια μιας στοίβας και αποθηκεύει τις αντιστοιχίες αριστερών και δεξιών παρενθέσεων σε μία FIFO ουρά, όπως αναφέραμε παραπάνω. Η `PrintOutput` τυπώνει στο ρεύμα εξόδου τις αντιστοιχίσεις των παρενθέσεων (δηλαδή τα περιεχόμενα της FIFO ουράς).

Εκτέλεση Προγράμματος

Εκτελέστε το συμπληρωμένο πρόγραμμα σας `Parentheses.java` με είσοδο κάθε φορά μία από τις παρακάτω ακολουθίες:

1. `[({ })] ()`
2. `[({ }) { [] () }] { [()] }`
3. `{ () }`
4. `) (`
5. `[()`
6. `[] ([`

Για παράδειγμα, η πρώτη ακολουθία είναι συντακτικά ορθή και το πρόγραμμα θα πρέπει να τυπώνει τα ζεύγη αριθμών: 2-3, 1-4, 0-5 και 6-7.

Για να εκτελέσετε το πρόγραμμα γράψτε

```
java Parentheses
```

Στη συνέχεια θα σας ζητηθεί να δώσετε μια έκφραση εισόδου, π.χ.

```
Enter input string > [ ( { } ) ] ( )
```

Σημείωση: Σε κάποια συστήματα μπορεί να χρειαστεί να δώσετε `Ctrl-D` μετά το `ENTER` για να διαβαστεί ο χαρακτήρας τερματισμού της εισόδου (`EOF`).

Αποθηκεύστε τα αποτελέσματα της κάθε εκτέλεσης στο αρχείο `lab3results.txt`.

Παραδοτέα

Ανεβάστε στο eCourse ένα zip αρχείο με τα τελικά σας προγράμματα `Parenthesis.java`, `Stack.java` και `Queue.java`, τα έτοιμα προγράμματα `In.java` και `Pair.java`, καθώς και με το αρχείο των αποτελεσμάτων `lab3results.txt`. Το zip αρχείο πρέπει να έχει το όνομα ενός μέλους της ομάδας σας με λατινικούς χαρακτήρες.