Object Oriented Programming Course

Activity No 2

RPN Calculator σε Java

Στόχος

Εξοικείωση με:

- Συγγραφή και εκτέλεση java εφαρμογής
- Κατασκευές της Java που η γλώσσα υιοθετεί από την C
- αξιοποίηση της βασικής βιβλιοθήκης της Java (Java API) https://docs.oracle.com/javase/8/docs/api/
- Δημιουργία project στο Eclipse

Η δραστηριότητα βασίζεται στην προηγούμενη (Δραστηριότητα No 1 – RPN Calculator σε C). Στην ουσία είναι μεταφορά της υλοποίησης με C της RPN Calculator της δραστηριότητας 1, στο περιβάλλον της Java με την ελάχιστη δυνατή αξιοποίηση των ΟΟ κατασκευών της γλώσσας.

Παρατήρηση: Μην διστάσετε να αξιοποιήσετε την επικοινωνία με τους συμφοιτητές σας και τους διδάσκοντες για να ξεπεράσετε τα δύσκολα στην κατανόηση ή εκτέλεση σημεία και πιθανόν να συμβάλετε στην βελτίωση της άσκησης. Καθώς δεν αναφέραμε ακόμη το Eclipse μπορείτε να δουλέψετε με το BlueJ.

Άσκηση. A Java implementation of the RPN Calculator

Η άσκηση έχει ως αντικείμενο την ανάπτυξη μιας Java εφαρμογής με την λειτουργικότητα της RPN αριθμομηχανής που αναπτύχθηκε στην δραστηριότητα No 1.

Σας δίνεται μέρος του πηγαίου κώδικα και σας ζητείται να συμπληρώσετε τον κώδικα ώστε να έχετε μια λειτουργούσα έκδοση της RPNCalculator.

Για απλοποίηση του κώδικα υποθέτουμε πως ο χρήστης δίνει την έκφραση παρεμβάλλοντας ανάμεσα σε τελεστές και τελεστέους ένα κενό χαρακτήρα, π.χ., "12 24 + =", "26 14 + 310 210 - + = " Υποστηρίζονται μόνο ακέραιοι τελεστέοι και οι τελεστές + - * / =.

Υποθέτουμε στη φάση αυτή ότι η έκφραση που έδωσε ο χρήστης ικανοποιεί τους παραπάνω περιορισμούς.

Σας ζητούνται τρεις εκδόσεις του προγράμματος με την σειρά που παρουσιάζονται παρακάτω.

1η έκδοση

Στην έκδοση αυτή το πρόγραμμα σας αποτελείται από μία μόνο κλάση, την `RpnCalculator`, μέρος της οποία σας δίνεται στο σχήμα 1.

Design Constraints:

use stack.

Implementation Constraints:

Υλοποιήστε την δική σας στοίβα.

Δεν θα ορίσετε άλλη κλάση εκτός από την `RpnCalculator`.

Μέθοδος main()

Η main καλεί την μέθοδο κλάσης getExpression() (θα πρέπει να δοθεί από εσάς) η οποία παίρνει από τον χρήστη την προς υπολογισμό έκφραση, π.χ. την 12 24 + = , την βάζει σε ένα στιγμιότυπο της String και επιστρέφει την αναφορά του στιγμιοτύπου αυτού, την οποία η main αναθέτει στην αναφορά exp που έχει δηλωθεί ως data member κλάσης (keyword static).

Αυτή η αναφορά θα χρησιμοποιηθεί στη συνέχεια από την getOp για να πάρει αυτή ένα-ένα τα συνθετικά της έκφρασης και να τα επιστρέψει στην main. Η main θα πρέπει να τα χειριστεί κατάλληλα, όπως έκανε και στην C υλοποίηση.

Μέθοδος getOp()

Όπως μπορείτε να παρατηρήσετε ακολουθείται η ίδια λογική με την getOp() της C υλοποίησης. Θα πρέπει να παίρνει από την προς υπολογισμό έκφραση, που θα έχει δώσει ο χρήστης, το επόμενο συνθετικό της και θα διαπιστώνει αν το συνθετικό αυτό είναι operand ή operator. Στην πληροφορία αυτή θα πρέπει στη συνέχεια να έχει πρόσβαση η main. Προς την κατεύθυνση αυτή έχουν οριστεί οι γνωστές σας

μεταβλητές input και inputType, οι οποίες έχουν οριστεί ως data members κλάσης (keyword static).

Προσέξτε πως η μέθοδος ορίζεται ως private static καθώς ορίζει συμπεριφορά κλάσης. Επιστρέφει δε int. Εναλλακτικά μπορεί να είναι και void.

```
public class RpnCalculator {
    static String exp;
    static String input;
    static int inputType;
    static Scanner sc;
    static int [] stack;
    static int sp=0;


    public static void main(String[] args) {
           // TODO Auto-generated method stub
           System.out.println("Reverse Polish Notation Calculator");
           stack = new int[20];
           exp= getExpression();

           sc = new Scanner(exp);
           sc.useDelimiter(" ");
           …
           }

    private static int getOp() {
           ….

           }
       // other class methods to be defined
       }
```

Σχήμα 1. Μέρος της κλάσης RpnCalculator της 1ης έκδοσης

Μέθοδος `getExpression()`

Αξιοποιήστε την κλάση Scanner της βασικής βιβλιοθήκης της Java για να πάρετε από την βασική είσοδο του συστήματος, στην οποία έχετε πρόσβαση με την αναφορά System.in, την προς υπολογισμό έκφραση. Αποθηκεύσετε την ως ένα στιγμιότυπο της String και επιστρέψετε την αναφορά του στην main.

Στοίβα

Στην έκδοση αυτή θα υλοποιήσετε μια στοίβα ακεραίων πρωτογενούς τύπου int. Προσέξτε την διαφορά (από την C) στον ορισμό του πίνακα stack. Η πρόταση

```
static int [] stack;
δηλώνει την stack ως αναφορά σε πίνακα ακεραίων. Η πρόταση της main
stack = new int[20];
```
δημιουργεί τον πίνακα των 10 ακεραίων.
Εσείς θα πρέπει να ορίσετε τις μεθόδους push and pop.

2η έκδοση – Ορισμός κλάσης MyStack

Χρησιμοποιήστε ως βάση την έκδοση 1.

Κλάση MyStack

Στην έκδοση 2 η στοίβα θα είναι στιγμιότυπο μιας δικής σας κλάσης MyStack την οποία θα πρέπει να ορίσετε και στη συνέχεια να τροποποιήσετε την RpnCalculator ώστε να την χρησιμοποιεί.

Ορίστε κατάλληλα την δομή της κλάσης σας (array of primitive int and sp index) και την συμπεριφορά της (push and pop methods)

3η έκδοση – Χρήση της κλάσης java.util.Stack

Χρησιμοποιήστε ως βάση την έκδοση 2.
Τροποποιήστε το πρόγραμμα σας ώστε να αξιοποιεί αντί για την δική σας κλάση `MyStack` την κλάση `java.util.Stack`.

Πρόσθετη προαιρετική λειτουργικότητα:

1. Τροποποιήστε το πρόγραμμα σας ώστε αυτό να δέχεται και να υπολογίζει τιμές εκφράσεων μέχρις ότου ο χρήστης δώσει 'q' ως ένδειξη τερματισμού προγράμματος.
2. Τροποποιήστε το πρόγραμμα σας ώστε να εκτελεί πράξεις με δεκαεξαδικούς αριθμούς.