
Evaluation d'une expression postfixe à l'aide d'une pile

Semestre d'hiver 2017/2018 – filière ITI

Objectifs

Ce laboratoire répond aux objectifs suivants:

- 1) Utilisation d'une pile pour le calcul d'une expression postfixe
- 2) Capture et traitement des exceptions
- 3) Boîtes de dialogue

Reddition

I. Descriptif

On implémente une calculatrice qui évalue les expressions postfixe avec une pile. Les exceptions dues à des erreurs de syntaxique ou de saisie seront capturées et traitées. Finalement, un affichage graphique à l'aide de boîtes de dialogues permettra de compléter le programme.

II. Enoncé

Le travail consiste à saisir une expression postfixe (comme p.ex. : "5 3 + 2.3 -") et à l'évaluer en utilisant la pile implémentée dans le TP précédent.

Le programme à réaliser («`Calc.java`») est une application autonome simple non fenêtrée pour obtenir le résultat arithmétique. Tout l'algorithme (une vingtaine de lignes) sera placé dans la fonction «`main(..)`»..

L'expression sera d'abord saisie dans la ligne de commande :

```
>java Calc 5 3 + 2.3 -
```

Le résultat (5.7) sera directement affiché dans la fenêtre de commande à l'aide de l'instruction `System.out.println(resultat)`.

Si, pour une raison ou une autre, l'expression saisie est incorrecte, votre programme lèvera une exception avec le message d'erreur "`Expression incorrecte`". Cette exception devra être traitée pour offrir à l'utilisateur la possibilité d'entrer à nouveau une expression.

Rappelons le principe d'évaluation d'une expression postfixe. Pour chaque terme de l'expression:

1. si c'est un nombre: l'empiler
2. si c'est un opérateur (+, -, *, /): dépiler les 2 derniers nombres, leur appliquer l'opérateur et empiler le résultat.

Indication : un nombre en `String` réussit sa conversion en un `Double`. Si la conversion échoue, on teste si la chaîne est faite d'un caractère qui soit un opérateur. Sinon il s'agit d'une erreur de saisie.

III. Implémentations

Etape A : architecture du programme

Programme du TP3

```
import java.util.ArrayList;

class Pile<E> {
    // votre implementation d'une pile générique du TP3
}

public class Calc {
    public static void main(String[] args) {
        Pile<Double> pileNb = new Pile<Double>();
        for (String t:args){
            // évaluation de l'expression postfixe
        }
        // afficher le résultat en mode texte
    }
}
```

QUESTIONS

1. Quelle est l'instruction qui lève l'exception `NumberFormatException` ?
2. Que se passe-t-il avec un argument `+ 5 3` ?
3. Que signifie la syntaxe de boucle `for (String t:args)` ?

Etape B : traiter les exceptions

Traiter les erreurs de saisie et de syntaxe en capturant et en traitant les exceptions correspondantes. Consultez l'API Java pour l'utilisation de la classe `StringTokenizer` qui vous facilitera le traitement des chaînes de caractères saisies au clavier (plutôt que passées en ligne de commande).

Etape C : affichage graphique

Inspirez-vous du programme suivant pour un simple affichage graphique.

Programme pour le GUI

```
import javax.swing.JOptionPane;

public class CalcGUI {
    public static void main(String[] args) {
        String phrase = JOptionPane.showInputDialog(
            "Entrer votre expression postfixe:"
        );
        double res = 0.0;
        // à compléter: calcul du <res> en évaluant l'expression postfixe
        // stockée dans le String <phrase>, à l'aide d'une pile
        JOptionPane.showMessageDialog(null, "Somme= " + res);
    }
}
```