

Travaux pratiques n° 2

Introduction à lex

Exercice 1 (Calculatrice en post-fixe)

Nous désirons écrire en *Lex* un programme permettant d'évaluer une expression post-fixée sur une ligne. Les différentes actions à réaliser sont les suivantes :

- Lorsqu'un entier est rencontré, il est placé dans une pile ;
- Lorsqu'un opérateur est rencontré ('-', '+', '*', '/'), deux entiers sont dépilés et le résultat est placé sur la pile.

Si la ligne se termine alors que la pile comporte plus d'un élément ou que le nombre d'entiers dans la pile est insuffisant pour réaliser une opération, une erreur est affichée.

- 1°) Proposez une solution pour gérer uniquement les entiers.
- 2°) Modifiez votre analyseur pour accepter des entiers négatifs.
- 3°) Que devrait-on modifier pour accepter des opérations de réels ?
- 4°) Modifiez votre programme pour que le retour à la ligne ne provoque plus d'erreur. Le caractère "=" affiche maintenant le contenu de la pile.
- 5°) Modifiez votre programme pour gérer les mots-clefs **SWAP** et **CLEAR** permettant respectivement d'inverser les deux entiers placés en haut de la pile et de vider la pile.

Exercice 2 (Détection d'entités)

Écrire un analyseur Lex permettant de reconnaître un ensemble d'entités : des nombres (entiers ou réels), un ensemble de mots-clefs (exemple : "debut" et "fin"), des opérateurs (-, +, * et /) et des chaînes de caractères (toute suite de caractères entre des "). À chaque fois qu'une entité est reconnue, l'analyseur l'affiche à l'écran. À la fin de l'exécution, l'analyseur indiquera le nombre de lignes et de chaque entité rencontrée.

Exemple : la chaîne "5-debut" générera "nombre 5 opérateur - mot-clef debut".

Exercice 3 (Java le dire à tout le monde)

Nous désirons écrire un analyseur de code Java.

- 1°) Écrire un analyseur Lex permettant de compter le nombre de blocs (entre accolades) et le nombre d'instructions (détectées par un point-virgule). Votre analyseur doit détecter si les blocs sont bien formés (*i.e.* le nombre d'accolades ouvrantes doit correspondre au nombre d'accolades fermantes).
- 2°) Modifiez l'analyseur pour qu'il indique pour chaque bloc, le numéro du bloc, la ligne de départ du bloc et le nombre d'instructions contenues dans le bloc.
- 3°) Écrire un analyseur Lex permettant d'extraire l'ensemble des commentaires Java d'un fichier passé en paramètre. Pour rappel, il y a deux types de commentaires : les commentaires sur plusieurs lignes entre /* et */ (qui ne peuvent être imbriqués) et sur une seule ligne précédés de //.
- 4°) Les questions 1) et 2) prennent-elles en compte les commentaires ? Les blocs en commentaires sont-ils ignorés ?