

## Projet INF404 : Interpréteur

### Introduction :

Notre programme est un interpréteur se basant sur les consignes données dans le TP5. Cette interpréteur permet d'effectuer des calculs et affectation à partir d'une expression donnée. Les résultat d'un calcul est lui stocké dans une variable.

### Fonctionnalités :

Notre interpréteur accueil le langage suivant :

- Les identificateurs (IDF) : Une lettre ou une suite de lettres.
- Le symbole d'affectation (AFF) : Le symbole '='.
- Le symbole séparant deux affectations (SEPAFF) : Le symbole ';'.
- Les nombres (ENTIER) : Une suite de chiffre (0 à 9) formante un nombre.
- Les symboles d'opérations (PLUS, MOINS, MUL, DIV) : Les symboles permettant d'effectuer des calculs entre 2 nombres. Les symboles disponibles sont les suivants : '+', '-', '\*', '/'.
- Les parenthèses (PARO et PARF) : permettent de créer une mécanique de priorité dans une expression, elles sont représentées par les symboles suivant : '(' et ')'.
- La fin de séquence (FIN\_SEQUENCE) : Est représentée par le symbole 'ε', elle désigne la fin de l'expression.

Grammaire du langage :

$$\begin{aligned}
 eag &\rightarrow seq\_terme \\
 seq\_terme &\rightarrow terme \text{ suite\_seq\_terme} \\
 suite\_seq\_terme &\rightarrow op1 \text{ terme suite\_seq\_terme} \\
 suite\_seq\_terme &\rightarrow \epsilon \\
 terme &\rightarrow seq\_facteur \\
 seqfacteur &\rightarrow facteur \text{ suite\_seq\_facteur} \\
 suite\_seq\_facteur &\rightarrow op2 \text{ facteur suite\_seq\_facteur} \\
 suite\_seq\_facteur &\rightarrow \epsilon \\
 facteur &\rightarrow ENTIER \\
 facteur &\rightarrow PARO \text{ eag PARF} \\
 facteur &\rightarrow IDF \\
 aff &\rightarrow IDF \text{ AFF eag SEPAFF} \\
 seq\_aff &\rightarrow aff \text{ seq\_aff} \\
 seq\_aff &\rightarrow \epsilon \\
 op1 &\rightarrow PLUS \\
 op1 &\rightarrow MOINS \\
 op2 &\rightarrow MUL \\
 op2 &\rightarrow DIV
 \end{aligned}$$

### Exemple et résultats :

Voici deux exemple d'exécution de notre programme avec leurs fichiers d'entrées respectifs et le résultat obtenue :

1<sup>ère</sup> exécution :

Fichier d'entrée (entree\_simple.txt) :

toto = 4 + 2;

Résultat de l'exécution :

```
Ast produit : (toto=(4+2))

-----
[AFFICHAGE DES VARIABLES]
-----

Position n°0 : Nom = toto, Valeur = 6.000000
-----
```

On observe bien que la variable avec comme nom 'toto' s'est vue affectée la valeur 6, résultat du calcul  $4 + 2$ .

2<sup>ème</sup> exécution :

Fichier d'entrée (entree\_final.txt) :

```
toto = 4 + 2;
y = 2 + 2 / 4;
z = 3 + y;
final = ( toto + y + z ) / 10;
```

Résultat de l'exécution :

```
Ast produit : ((toto=(4+2))
((y=(2+(2/4)))
((z=(3+y))
(final=((toto+y)+z)/10))))

-----
[AFFICHAGE DES VARIABLES]
-----

Position n°0 : Nom = toto, Valeur = 6.000000
Position n°1 : Nom = y, Valeur = 2.500000
Position n°2 : Nom = z, Valeur = 5.500000
Position n°3 : Nom = final, Valeur = 1.400000
-----
```

Cette fois si on observe bien l'affectation des calculs aux variables 'toto', 'y', 'z', et 'final'. La variable final a la particularité d'effectuer un calcul à partir des 3 variables affectées précédemment. Les valeurs des 3 variables 'toto', 'y' et 'z' additionnées entre elles donnent le résultat 14, qui une fois diviser par 10 donne bien le résultat 1,4 qui apparaît comme valeur de la variable 'final'.