

### Projet mini langage

#### Informations pratiques :

- Le projet est à réaliser en binôme
- Les soutenances auront lieu le 5 juillet 2017
- Présoutenance le 13 juin

#### Sujet :

L'objectif du projet est de concevoir un interpréteur pour un mini langage. Il est obligatoire de baser l'interprétation sur un arbre de syntaxe abstrait construit au cours de l'analyse syntaxique du « programme » donné en entrée.

Le projet doit vous permettre d'acquérir les compétences suivantes :

- générer des analyseurs lexicaux et syntaxiques avec lex et yacc
- construire un arbre de syntaxe abstrait à partir d'un arbre concret
- interpréter la chaîne d'entrée (le programme) en évaluant l'arbre de syntaxe abstrait correspondant.

Spécifications de la version minimale (11/20) :

1. Votre interpréteur devra gérer les noms de variables à plusieurs caractères.
2. Prendre en compte le type réel
3. Gérer les instructions suivantes :
  - a. affectation
  - b. affichage d'expressions numériques (pouvant contenir des variables numériques)
  - c. instructions conditionnelles : implémenter le si-alors-sinon/si-alors
  - d. structures itératives : implémenter le while et le *for*
  - e. Affichage de l'arbre de syntaxe et de la table des variables

#### Améliorations possibles (entre autre):

1. Gestion des erreurs (variable non initialisée, ...)
2. Gestion de la saisie clavier
3. Gestion du type chaîne de caractères (et extension d'autant de l'instruction d'affichage)
4. Gérer les fonctions
5. Gérer la déclaration explicite des variables
6. Gérer la portée des variables et les fonctions récursives
7. Les pointeurs
8. Les tableaux
9. la POO

Exemple de code (la syntaxe du langage ainsi que les mots clefs peuvent être librement choisis):

```
12;  
AFFICHER(12);  
AFFICHER(12 + 8);  
x = 0;  
SI x == 0 ALORS AFFICHER(1); SINON AFFICHER(0); FIN;  
fm1 = 1;  
fm2 = 1;  
ftmp = 0;  
TANTQUE x < 20 FAIRE x = x + 1; ftmp = fm1 + fm2; fm2 = fm1; fm1 = ftmp; AFFICHER(ftmp);  
FIN;
```

Exemple de sortie :

```
12.000000  
12.000000  
20.000000  
x <- 0.000000  
1.000000  
- Fait -  
fm1 <- 1.000000  
fm2 <- 1.000000  
tmp <- 0.000000  
2.000000  
3.000000  
5.000000  
8.000000  
13.000000  
21.000000  
34.000000  
55.000000  
89.000000  
144.000000  
233.000000  
377.000000  
610.000000  
987.000000  
1597.000000  
2584.000000  
4181.000000  
6765.000000  
10946.000000  
17711.000000  
- Fait -
```