

### Projet mini langage

#### Informations pratiques :

- Le projet est à réaliser en binôme
- Une présoutenance aura lieu lors du dernier cours avant la soutenance
- Le projet est à rendre par email au plus tard le dimanche 27 avril
  - o à [fbaudoin@my-itcampus.com](mailto:fbaudoin@my-itcampus.com)
  - o Objet : [TDL PROJET] NOM1 NOM2
  - o PJ :
    - sources zippées
    - bref rapport décrivant le périmètre des fonctionnalités implémentées + les éventuels bugs connus
- Les soutenances auront lieu le 30 avril.

#### Sujet :

L'objectif du projet est de concevoir un interpréteur pour un mini langage. Contrainte : l'interprétation doit se baser sur un arbre de syntaxe construit au cours de l'analyse syntaxique de la suite d'instructions en entrée. (et pas sur une interprétation à la volée au cours du parsing)

#### Spécifications de la version minimale (12/20) :

1. Votre interpréteur devra gérer les noms de variables à plusieurs caractères.
2. Prendre en compte les types réel et booléen (explicitement ou non)
3. Gérer les instructions suivantes :
  - a. affectation
  - b. affichage : implémenter l'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

#### Améliorations possibles :

1. Affichage de l'arbre de syntaxe et de la table des variables
2. Gestion des boucles *pour*
3. Gestion du type chaîne de caractères (et extension d'autant de l'instruction d'affichage)
4. Gérer les fonctions
5. Gérer la portée des variables et les fonctions récursives
6. Version interactive : l'utilisateur entre les instructions qui sont interprétées les unes après les autres.
7. Version interactive avec récupération d'erreur : En cas d'erreur de syntaxe, on revient à l'instruction d'avant (cf setjmp.h) : on ne redémarre pas le parsing au début.

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 -
```