

## NF11 – TP5 : GENERATION D'ANALYSEUR LEXICAL ET SYNTAXIQUE (4EME SEANCE)

---

### ETAPE 4 - 5 : PROCEDURES ET FONCTIONS

---

On considèrera que toutes les procédures et les fonctions doivent être déclarées avant le programme principal et donc avant leur appel.

---

#### REPRESENTER

---

- Modifier le visiteur pour préparer la pile d'exécution.
- Les procédures sans et avec paramètres.

On sera amené à créer une classe `Procedure` pour conserver lors du parsing de sa déclaration tous les attributs nécessaires à l'exécution d'une procédure.

Toute variable utilisée dans une procédure sera considérée locale à la procédure. Elle masquera lors de l'exécution de la procédure une variable de même nom du programme principal et de tout autre procédure active (une procédure peut en appeler une autre).

Quel est le type de passage de paramètres que vous envisagez ?

Exemple :

```
pour petitcarre :n
  donne "cote :n / 2
  repete 4 [ av :cote td 90 ]
fin
tracecarre 100 donne "cote 200
tracecarre :cote
```

- Les fonctions sans et avec paramètres. Les procédures et les fonctions sont appelées différemment. Quel est le problème spécifique des fonctions ?

Exemple :

```
pour carre :c
  rends :c * :c
fin
av carre 10
```

On considèrera que l'instruction `rends` est la dernière précédant `fin`. Si ce n'était pas le cas il faudrait pouvoir arrêter l'exécution des listes d'instructions (voir également `stop`) d'une procédure.

Exemple :

<pre>pour doublesi :n   si :n &gt;= 100 [rends 2 * :n]   rends :n fin</pre>	<pre>pour doublesi :n   donne "t :n   si :n &gt;= 100 [donne "t 2 * :n]   rends :t fin</pre>
---	--

Dans cet exemple, `doublesi 50` rends 50 et `doublesi 100` rends 200.

Analysez des instructions comme `av double 90 + 20`. Elle peut a priori se réécrire comme `av 110` ou comme `av 220`.

- Les procédures récursives. Vérifier si la modélisation actuelle vous permet de créer des procédures récursives.