Université des Sciences et de la Technologie d'Oran Mohamed BOUDIAF
FACULTE DES MATHEMATIQUES ET INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

TP – COMPILATION – 2021/2022

Microprojet en TP COMPILATION

Dans la première partie du TP en Compilation, nous avons parlé des trois types d'analyse (Lexicale, syntaxique, et sémantique) puis abordé l'analyse lexicale. Nous avons vu comment générer un analyseur lexical en utilisant le langage **LEX**. Ainsi, nous avons pu **identifier** chaque expression suivant des **règles** définies, par exemple, pour une expression régulière qui **définissait un mot de longueur 3**, nous avons pris l'expression **LEX : [A-Z|a-z|0-9]{3}** soit une chaîne composée de trois caractères dans les ensembles des lettres minuscules, majuscules ou des chiffres et d'**aucun autre type de caractère**.

En parallèle aux exercices sur les différents types d'analyses, ce microprojet à réaliser au cours du semestre, doit pouvoir analyser un fichier source écrit dans le langage donné, en suivant les trois étapes d'analyse citées précédemment.

Langage de programmation et contraintes : C++ / Java (Builder /eclipse/netbeans),

Toutes les fonctions/méthodes prédéfinies du langage choisi sont autorisées **sauf celle qui ont un rapport avec l'analyse partielle ou totale d'une chaine de caractère quel que soit le type d'analyse**. Vous êtes bien-évidemment autorisés à programmer vous-même des fonctions qui facilite l'analyse, par exemple : une fonction qui prend en entrée un caractère et qui retourne son type (Lettre, chiffre, autres, etc.). L'application doit être présentée avec une interface (une fenêtre visuelle) avec au moins un bouton pour rechercher un fichier et trois autres correspondants à chaque type d'analyse. Nous supposerons que notre langage s'appelle « **SNAIL** » et nos fichiers sources auront l'extension « .snl ».

Spécification du langage : le code source doit être écrit de cette façon :

Une et une seule instruction par ligne.
Le programme commence toujours par « **Snl_Start** » et se termine par « **Snl_Close** »
Un commentaire prend une ligne entière et doit commencer par « **%..** »
Une instruction finit toujours par « **%.** »
Une chaîne de caractère est, comme pour le langage C mise entre guillemets **mais** ne contient que des lettres, des chiffres, ou des apostrophes, et déclarée avec le mot clé « **SnlSt** ».
Les expressions régulières correspondantes aux types de variables reconnus par le langage sont :

Type de variable	Description	Expression régulière en LEX
Un identificateur	Décrit dans la fiche 1 – dernière question	{Lettre}(_?({Lettre}|{Chiffre})+)*
Un nombre entier	composée d'au moins un chiffre	[0-9]+
Un nombre réel	Composé de deux entiers séparés par un point	[0-9]+\.[0-9]+

En plus des variables, le langage accepte aussi les commandes suivantes :

Commande	Description	Exemple
Affection d'un nombre entier ou réel à une variable	Set « identificateur » « valeur » %.	Set j 55 %.
Affectation de la valeur d'une variable à une autre variable	Get « identificateur » from « identificateur » %.	Get j from i %.
Condition	If % « condition » % do « action » %.	If % i<j % do Set j 6 %.

Université des Sciences et de la Technologie d'Oran Mohamed BOUDIAF
FACULTE DES MATHEMATIQUES ET INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

TP – COMPILATION – **2021- 2022**

Pour le reste des contraintes, le tableau suivant donne un exemple d'un programme écrit dans ce langage :

Syntaxe d'écriture des instructions	**Définition sémantique**
`Snl_Start`	Début du programme
`Snl_Int i,j,Aft_5,f_f_5 %.`	Déclaration de quatre entiers
`Set i 23 %.`	Affectation d'une valeur à i
`Snl_Real Aft34_2 %.`	Déclaration d'un réel
`If % i<j %`	Condition
`Set Aft_5 10 %.`	Action
`Else`	Sinon
`  Start`	Début d'un bloc
`    Get j from i %.`	Affectation de valeur entre variables
`    Set Af34_2  123,54 %.`	Affectation d'une valeur réelle à Af34_2
`  Finish`	Fin d'un bloc
`Snl_Put " ceci est un message "%.`	Affichage d'un message à l'écran
`Snl_Put i %.`	Affichage de la valeur de i
`%.. ceci est un commentaire`	Exemple d'un commentaire
`Snl_Close`	Fin du programme

Exemple d'interface :

La figure suivante présente un exemple de l'interface attendue avec l'exécution de l'analyse lexicale :

Le fichier contient notre code source à analyser, il est chargé dans l'application et les différents types d'analyses sont lancés.