NF11 – TP-PRODUCTION-CODE : PRODUCTION DE CODE A PARTIR D'UNE GRAMMAIRE

OBJECTIFS

Créer une grammaire d'un langage PreLogo

Générer un analyseur syntaxique de cette grammaire.

Créer un générateur de programme en langage Logo.

OUTILS

Projet Eclipse contenant les éléments à compléter : fichier de template, grammaire, visiteur d'arbre.

INTRODUCTION

On considère une grammaire permettant d'écrire des fichiers Pré-logo, destinés à être source de programmes Logo, et ayant la forme :

```
import pointh
import pointb
import rectangle
pointh(0 0)
pointb(100 100 #3)
pointb(100 0 #6)
pointb(0 0 #8)
pointh(100 100 #5)
rectangle(200 50 #1) nb = 36
```

Un tel fichier comporte une liste d'import suivi d'une liste de commandes reprenant les noms des import. Les commandes ont deux paramètres entiers et un troisième paramètre facultatif de la forme # suivi d'un chiffre. Une commande est suivie d'un paramètre nb facultatif entier indiquant un facteur de répétition. Le fichier précédent est destiné à produire du code Logo de la forme :

```
pour pointh :x :y                    fcc 0
  lc                                 pointh( 0 0 )
  fpos [ :x :y ]                     fcc 3
  bc                                 pointb( 100 100 )
fin                                  fcc 6
pour pointb :x :y                    pointb( 100 0 )
  fpos [ :x :y ]                     fcc 8
fin                                  pointb( 0 0 )
pour rectangle :long :larg           fcc 5
  av :long td 90 av :larg td 90      pointh( 100 100 )
  av :long td 90 av :larg td 90      fcc 1
fin                                  repete 36 [rectangle( 200 50 ) td 360 / 36]
```

Les import permettent d'inclure des procédures à deux paramètres. Les commandes permettent de générer du code Logo. Chaque commande se traduit par l'appel à une procédure importée

avec les deux paramètres entiers. Le paramètre facultatif désigne la couleur du tracé (qu'il faut définir au préalable par fcc – 0 par défaut). Le dernier nombre facultatif définit le facteur de répétition (1 par défaut). La forme est alors répétée autant de fois avec une rotation sur le point origine de façon à accomplir un tour.
On ne prendra pas en compte les erreurs possibles lors de l'écriture du fichier prelogo.txt, telles que l'appel à un import qui n'aurait pas été fait.

QUESTIONS

1. Compléter la grammaire PreLogo pour qu'elle puisse décrire les fichiers Pre-Logo. Générer le code du parseur. Tester avec le fichier exemple précédent.
2. Compléter le fichier de templates (templates/prelogo.stg) et le générateur (prelogoparsing/LogoGenerator) pour que la sortie console du générateur contienne les définitions des procédures. On remarquera que ce sont les rendus des templates qui sont inclus dans le fichier de sortie.
3. Compléter le fichier de templates et le générateur pour générer l'écriture des commandes. On pourra être amené à créer une classe Java. Dans un premier temps on considèrera que l'écriture comportera systématiquement une instruction Logo repete même si le facteur de répétition est de 1 :

```
repete 1 [pointh( 100 100 ) td 360 / 1]
```

4. Tester le résultat dans l'interpréteur graphique du Logo.
5. Amélioration : le moteur de StringTemplate comporte une inclusion conditionnelle portant sur l'existence de la valeur d'un attribut ou d'un champ d'un attribut si celui-ci est un objet. Ainsi on peut insérer dans un template une écriture telle que :

```
$if(x)$$x:more()$        ou    $if(com.y)$$com:more()$
$endif$                        $else$$com:once()$
                               $endif$
```

   Elle signifie : si x est non null ecrire x avec le template more (qu'il faut écrire dans le fichier de template). La partie droite signifie : Si le champ y de l'attribut com est non null écrire l'attribut com avec le template more sinon l'écrire avec le template once. Si l'on considère que le paramètre nb, facteur de répétition est null par défaut au lieu de valoir 1 on peut utiliser la forme de droite précédente et obtenir des lignes telles que :

```
        pointh( 100 100 )
ou      repete 36 [rectangle( 200 50 ) td 360 / 36]
```