

NF11 – TP3 : GENERATION D'ANALYSEUR LEXICAL ET SYNTAXIQUE (2EME SEANCE)

ETAPE 2 : EXPRESSIONS ARITHMETIQUES

De façon cyclique :

ajouter une règle à la grammaire du langage LOGO ou réorganiser les règles afin d'ajouter une nouvelle fonctionnalité.

Faire générer les classes.

Compléter les classes LogoTreeVisitor et Traceur. Tester.

TESTER chaque ajout un à un.

REPRESENTER

- Ajouter dans la grammaire les règles permettant d'écrire une expression arithmétique utilisant les opérateurs + et - et les parenthèses. La justification de l'écriture des règles pour les expressions arithmétiques sera donnée en cours. Les valeurs numériques seront a priori indiquées en entier mais on pourra les calculer en flottant.
- Réorganiser la grammaire pour utiliser les expressions arithmétiques dans les instructions d'un programme. Ex : av 100 - (10 + 20)
- Modifier le visiteur d'arbre pour qu'il puisse interpréter correctement les expressions arithmétiques.
- A quoi sert le champ `atts` de type `ParseTreeProperty<Integer>` ?
- Ajouter la fonction hasard prenant en paramètre un entier.
Exemple d'utilisation : av hasard 100.
- Ajouter la répétitive : repete.
Ex : repete 4 [av 100 td 90]
- La variable loop.
ex : repete 4 [av 50 * loop td 90]

A RENDRE SUR MOODLE

La grammaire Logo.g4 dans son état actuel.

Ajouter un commentaire à la grammaire répondant à la question :

- Comment sont interprétés : av hasard 200 + 100 et av 200 + hasard 100. Pourquoi ?