

Projet de compilation

Binôme : Sagno Félix & Cissé Mamadou

Fonctionnalités Disponible de notre projet de compilation

Déclaration de variable,
Affectation des valeurs aux variables
Affichage des variables
Utilisation des conditions
Comparaison de variables dans une condition
Lecture d'une variable
Les commentaires sur une ligne

Règles de prduction grammaticale :

%start program

%%

program : listRetour

 |
 listDeclaration GUI INLINE listInstruction NEA listRetour
 ;

listDeclaration : declaration

 |
 listDeclaration declaration
 ;

declaration : IDENTIF INLINE;

listInstruction :instruction

 |
 listInstruction instruction
 ;

instruction : affichage

 |
 affectation
 |
 condition

```

        |
        lecture
        |
        commentaire
        ;
commentaire :    COMMENT CHAINE INLINE
                ;
lecture :        READ CRO_O expression CRO_F INLINE
                ;
affichage :     WRITE CRO_O contenu CRO_F INLINE
                ;
contenu: simple_affichage
        |
        texte
        |
        concat
        ;
concat: texte POINT simple_affichage
        |
        simple_affichage POINT texte
        |
        texte POINT simple_affichage POINT texte
        ;
texte: GRIF list_chaine GRIF
      ;
list_chaine: CHAINE
            |
            list_chaine CHAINE
            ;
simple_affichage: ENTIER
                |
                IDENTIF
                ;

affectation : IDENTIF EGAL expression INLINE
              ;
condition:    condition_si POINT INLINE
            |
            condition_si POINT INLINE condition_sinon
            |

```

```

        condition_si POINT INLINE list_condition_sinon_si
        ;
list_condition_sinon_si: condition_sinon_si POINT INLINE
        |
        condition_sinon_si POINT INLINE
condition_sinon
        |
        list_condition_sinon_si
condition_sinon_si POINT INLINE
        ;
condition_sinon_si: SINON SI CRO_O comparaison CRO_F INLINE
listInstruction
        ;
condition_si : SI CRO_O comparaison CRO_F INLINE listInstruction
        ;
condition_sinon: SINON INLINE listInstruction POINT INLINE
        ;

comparaison: IDENTIF signe_comparaison expression
        ;
signe_comparaison: EGAL CONDI
        |
        SUPEG
        |
        INFEG
        |
        INF
        |
        SUP
        |
        DIFF
        ;
expression : IDENTIF
        |
        ENTIER
        ;
listRetour: Retour
        |
        listRetour Retour
        ;

```

Retour : INLINE;
%%

Exemple de programme source :

```
@val1
@val2
Gui
    write["Debut du Programme"]
    @val=10
    @val2=09
    read[@val2]
    #[@val1==10]
        @val=20
    .
    //un commentaire ici
    write["fin du Programme"]
nea
```