**Grammaire**

S’ := PROG $

PROG := LOBJETS BLOC

LOBJETS := ε

| OBJET LOBJETS

OBJET := CLASSE

| OBJETISOLE

CLASSE := class Ident (OPTLPARAM) OPTHERITAGE OPTBLOC is { LCHAMP LMETHODE }

OPTHERITAGE := ε

| extends Ident (OPTLPARAM)

OBJETISOLE := object Ident OPTBLOC is {LCHAMP }

PARAM := Ident TYPE

OPTTYPE := ε

| TYPE

TYPE := : Ident

OPTLPARAM := ε

| LPARAM

LPARAM := PARAM

| PARAM , LPARAM

LCHAMP := ε

| CHAMP LCHAMP

CHAMP:= var OPTAUTO PARAM ;

OPTAUTO := ε

| auto

METHODE := def OPTOVERRIDE Ident (OPTLPARAM) METHODECORPS

METHODECORPS := TYPE := EXPR

| OPTTYPE is BLOC

OPTOVERRIDE := ε

| override

LMETHODE := ε

| METHODE LMETHODE

OPTBLOC := ε

| BLOC

BLOC := { OPTLINSTRUC }

| { LDECLVAR is LINSTRUC }

OPTLINSTRUC := ε

| LINSTRUC

LINSTRUC := INSTRUC

| INSTRUC LINSTRUC

LDECLVAR := DECLVAR

| DECLVAR OPTLDECLVAR

DECLVAR := Ident : TYPE ;

EXPR :=

**Token :**

ID CSTE

AFFECT

ET OU

PARENT\_G PARENT\_D POINTVIRGULE DEUXPOINTS POINT

IF THEN ELSE

ACCOLADE\_G ACCOLADE\_D

IS VAR

AUTO OBJECT DEF NEW