

Projet compilation - Grammaire

$\text{Prog} ::= \text{LOptDef} \{ \text{OptBloc} \}$

$\text{LOptDef} ::= \epsilon \mid \text{Def LOptDef}$

$\text{Def} ::= \text{object Id is} \{ \text{LOptDecl} \} \mid \text{class Id (LOptParamClasse) OptExtends is} \{ \text{BlocClasse} \}$

$\text{LOptDecl} ::= \epsilon \mid \text{Decl LOptDecl}$

$\text{Decl} ::= \text{var Id : Id OptAffectExpr ;} \mid \text{def OptOverride Id (LOptParamMethode) FinDeclMethode}$

$\text{OptAffectExpr} ::= \epsilon \mid := \text{Expr}$

$\text{OptOverride} ::= \epsilon \mid \text{override}$

$\text{LOptParamMethode} ::= \epsilon \mid \text{LParamMethode}$

$\text{LParamMethode} ::= \text{Id : Id} \mid \text{Id : Id , LParamMethode}$

$\text{FinDeclMethode} ::= : \text{Id} := \text{Expr} \mid \text{OptTypeRetour is} \{ \text{OptBloc} \}$

$\text{OptTypeRetour} ::= \epsilon \mid : \text{Id}$

$\text{OptBloc} ::= \epsilon \mid \text{LInstr} \mid \text{LDeclBloc is LInstr}$

$\text{LInstr} ::= \text{Instr} \mid \text{Instr LInstr}$

$\text{LDeclBloc} ::= \text{Id : Id OptAffectExpr ;} \mid \text{Id : Id OptAffectExpr ; LDeclBloc}$

$\text{LOptParamClasse} ::= \epsilon \mid \text{LParamClasse}$

$\text{LParamClasse} ::= \text{OptVar Id : Id} \mid \text{OptVar Id : Id , LParamClasse}$

$\text{OptVar} ::= \epsilon \mid \text{var}$

$\text{OptExtends} ::= \epsilon \mid \text{extends Id}$

$\text{BlocClasse} ::= \text{LOptDecl def Id (LOptParamClasse) OptSuper is} \{ \text{OptBlocConstr} \} \text{ LOptDecl}$

$\text{OptSuper} ::= \epsilon \mid : \text{Id (LOptArg)}$

$\text{LOptArg} ::= \epsilon \mid \text{LArg}$

$LArg ::= Expr \mid Expr, LArg$

$OptBlocConstr ::= \epsilon \mid LDeclBloc\ OptBlocConstr \mid Instr\ OptBlocConstr$

$Instr ::= Expr ; \mid OptBloc \mid \textbf{return}\ OptExpr ; \mid Cible := Expr ; \mid \textbf{if}\ Expr\ \textbf{then}\ Instr\ \textbf{else}\ Instr$

$OptExpr ::= \epsilon \mid Expr$

$Cible ::= Id \mid (Cible) \mid (\text{as}\ Id : Cible) \mid Cible\ LId \mid AppelFonction\ LId$

$LId ::= .Id \mid .Id\ LId$

$Expr ::= Id \mid Cste \mid (Expr) \mid (\text{as}\ Id : Expr) \mid Selection \mid new\ Id\ (LOptArg) \mid AppelFonction \mid ExprOperator$

$ExprOperator ::= + \mid - \mid * \mid /$

$Selection ::= Expr . Id$ // A exclure les Expr qui ne marchent pas ici dans la vérification contextuelle

$AppelFonction ::= Expr . Id\ (LOptArg)$