SBFSBF

⟨*Ident*⟩`e
SBF\⟨*Sep*⟩ ::= `;`|⟨*NewLine*⟩
\è⟨*NewLine*⟩
ò
o
è
⟨*Literal*⟩ ::= ⟨*Int*⟩
⟨*Float*⟩
⟨*Char*⟩
⟨*Boolean*⟩
⟨*String*⟩
⟩\
`/**//*commento*/*/`è
SBF
*programma*è⟨*Sep*⟩
dichiarazione
Dichiarazione di variabili e valori.⟨*Decl*⟩ ::= `val`⟨*Ident*⟩`:`⟨*TypeSpec*⟩`=`⟨*Expr*⟩
|`val`⟨*Ident*⟩`:`⟨*TypeSpec*⟩
|`var`⟨*Ident*⟩`:`⟨*TypeSpec*⟩`=`⟨*Expr*⟩
|`var`⟨*Ident*⟩`:`⟨*TypeSpec*⟩⟨*TypeSpec*⟩ ::= ⟨*SimpleType*⟩
|`&`⟨*TypeSpec*⟩
|`Array[`⟨*TypeSpec*⟩`]`
⟨*SimpleType*⟩ ::= `bool`|`char`|`integer`|`float`|`string`
Dichiarazione di funzioni/procedure.⟨*Decl*⟩ ::= `def`⟨*Ident*⟩⟨*ParamClauses*⟩`:`⟨*TypeSpec*⟩`=`⟨*Expr*⟩
`def`⟨*Ident*⟩⟨*ParamClauses*⟩`:`⟨*TypeSpec*⟩`=`⟨*Block*⟩⟨*ParamClauses*⟩è⟨*ParamClause*⟩⟨*ParamClause*⟩ ::= `(`⟨*ListOfParameters*⟩
*blocco*è⟨*Sep*⟩
istruzione⟨*Stmt*⟩ ::= `if(`⟨*Expr*⟩`)`⟨*Stmt*⟩
|`if(`⟨*Expr*⟩`)`⟨*Stmt*⟩`else`⟨*Stmt*⟩
|`if(`⟨*Expr*⟩`)`⟨*Stmt*⟩⟨*Sep*⟩`else`⟨*Stmt*⟩
|`while(`⟨*Expr*⟩`)`⟨*Stmt*⟩
|`do`⟨*Stmt*⟩`while(`⟨*Expr*⟩`)`
|`do`⟨*Stmt*⟩⟨*Sep*⟩`while(`⟨*Expr*⟩`)`
|`return`
|`return`⟨*Expr*⟩
|⟨*Block*⟩
|⟨*LExpr*⟩
|⟨*LExpr*⟩⟨*AssignmentOp*⟩⟨*Expr*⟩
|⟨*Ident*⟩⟨*ParamClauses*⟩
left expressions⟨*LExpr*⟩ ::= ⟨*Ident*⟩
|⟨*LExpr*⟩`(`⟨*Expr*⟩`)`
|`*`⟨*LExpr*⟩
|`&`⟨*LExpr*⟩
|`(`⟨*LExpr*⟩`)`
right expressions⟨*Expr*⟩ ::= ⟨*LExpr*⟩
|⟨*Literal*⟩
|⟨*Expr*⟩⟨*BinOp*⟩⟨*Expr*⟩
|⟨*UnOp*⟩⟨*Expr*⟩
|⟨*Ident*⟩⟨*Lists*⟩
|`(`⟨*Expr*⟩`)`⟨*BinOp*⟩ ::= `||`|`&&`|`+`|`-`|`*`|`/`|`%`|`^`|`==`|`!=`|`<`|`<=`|`>`|`>=`
⟨*UnOp*⟩ ::= `!-`