

Specifiche Lessicali

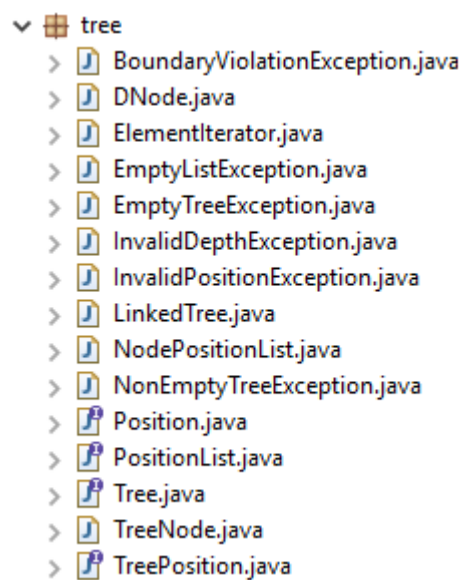
La grammatica utilizzata è esattamente quella fornita sulla piattaforma e-learning, così come la specifica lessicale a cui sono stati aggiunte la definizione regolare dei commenti.

Parser – CUP

Le azioni semantiche associate alle produzioni della grammatica permettono di costruire un albero sintattico.

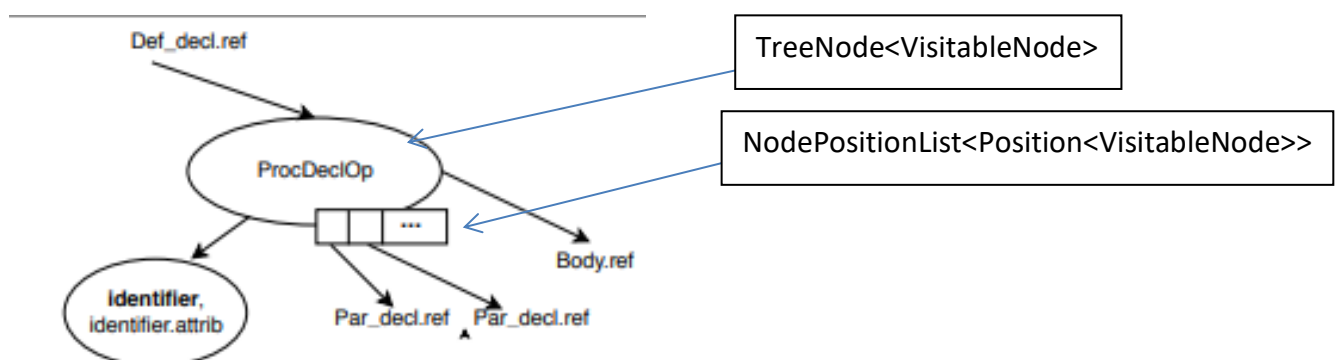
Per costruire tale albero sono state rispettate le specifiche date.

Le strutture dati implementate per costruire l'albero sintattico sono le seguenti:



Ciascun non terminale è un oggetto **TreeNode<VisitableNode>** oppure **NodePositionList<Position<VisitableNode>>** utili a definire l'oggetto **LinkedTree<VisitableNode>**, ovvero l'albero sintattico.

Es:



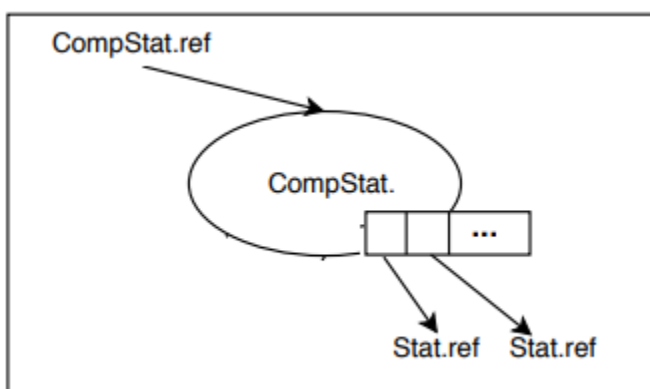
La classe **VisitableNode<E>** è contiene:

- Label – nome del nodo
- Riferimento alla tabella dei simboli – per la gestione dello scoping
- Tipo/TipoSpeciale - per la gestione del type checking

Sono state definite varie sottoclassi di **VisitableNode**, ognuna per marcare un determinato ruolo del tipo di nodo:

- **ScopingNode** – Nodo interno: marca un nuovo scope
- **TypedNode** – Nodo interno: marca un nodo tipato;
- **ExpressionNode** – Nodo interno: marca espressione aritmetiche, relazionale, booleana, unaria;
- **FuncDeclarationNode** – Nodo interno: marca un nuovo scope e dichiarazione di funzione
- **VarDeclarationNode** – Nodo interno: marca una dichiarazione di variabile
- **ParDeclarationNode** – Nodo interno: marca una dichiarazione di un parametro di una funzione
- **Leaf** – Nodo foglia: marca una foglia dell'albero;
- **ConstNode** – Nodo foglia: marca una costante
- **IDNode** – Nodo foglia: marca un identificatore
- **DeclarationIDNode** – Nodo foglia: marca una dichiarazione di un identificatore
- **UsageIDNode** – Nodo foglia marca un utilizzo di un identificatore

Compstat Nodo



Syntax Tree XML, Scoping, Type Checking e Generazione Codice C

Il risultato finale è un oggetto **LinkedTree<VisitableNode>** su cui si possono eseguire varie visite (preorder, postorder) per determinati scopi.

- **PrintXMLTree**: crea un file syntaxTree.xml che rappresenta l'albero sintattico;
- **ScopingTree**: definisce una tabella dei simboli ad ogni nuovo scope, la cui gestione è implementata tramite uno stack; controllo di dichiarazioni multiple o utilizzi di variabili non dichiarate; associa ad ogni nodo il riferimento alla relativa tabella dei simboli e il tipo;
- **GenerationSourceC**: : prende in input un file scritto in YASPL, lo compila, e restituisce un file C (source.c), effettuando una visita dell'albero.
- **TypeCheckingTree**: controllo di tipi in accordo al TypeSystem.

Oltre alle specifiche date, in aggiunta:

AddOp, DiffOp, MulOp, DivOp

	Integer	Double	String	Char	Bool
Integer	Integer	Double	Error	Error	Error
Double	Double	Double	Error	Error	Error
String	Error	Error	Error	Error	Error
Char	Error	Error	Error	Error	Error
Bool	Error	Error	Error	Error	Error

GTOP, GOp, LTOP, LEOp

	Integer	Double	String	Char	Bool
Integer	Bool	Bool	Error	Error	Error
Double	Bool	Bool	Error	Error	Error
String	Error	Error	Error	Error	Error
Char	Error	Error	Error	Error	Error
Bool	Error	Error	Error	Error	Error

EQOp

	Integer	Double	String	Char	Bool
Integer	Bool	Bool	Error	Error	Error
Double	Bool	Bool	Error	Error	Error
String	Error	Error	Bool	Error	Error
Char	Error	Error	Error	Bool	Error
Bool	Error	Error	Error	Error	Bool

AndOp, OrOp

	Integer	Double	String	Char	Bool
Integer	Error	Error	Error	Error	Error
Double	Error	Error	Error	Error	Error
String	Error	Error	Error	Error	Error
Char	Error	Error	Error	Error	Error
Bool	Error	Error	Error	Error	Bool

AssignOp

	Integer	Double	String	Char	Bool
Integer	Integer	Integer	Error	Error	Error
Double	Double	Double	Error	Error	Error
String	Error	Error	String	String	Error
Char	Error	Error	Error	Char	Error
Bool	Error	Error	Error	Error	Bool

UMinusOp

Tipo -> Integer, Double

CallOp

Tipo -> insieme dei tipi dei parametri

Controllo:

- Numero di parametri chiamata a funzione
- Tipo di parametri passati alla funzione
- Uso dei parametri all'interno della funzione (IN,OUT,INOUT)

WriteOp e ReadOp

Tipo -> insieme dei tipi dei parametri