

أمثلة المحلل اللغوي (Parser Examples)

مراجعة البنية النحوية



قواعد النحو (Grammar)



هذا يعني أنـ الـ (Lexer) التعليقات (التي تبدأ بـ // وتستمر حتى نهاية السطر) يتم التعامل معها وتتجاهلها بالكامل بواسطة المحلـ اللغويـ تمثلـ التعليقات، وبالتالي يمكن كتابةـ التعليقاتـ فيـ أيـ مكانـ فيـ الكـودـ دونـ التـأثيرـ عـلـىـ بنـيةـ الجـملـةـ النـحوـيةـ (Tokens)ـ لـنـ يـتـلقـىـ أيـ رـمـوزـ Parserـ.

◆ القواعد النحوية الأساسية

البرنامج (Program)

Program → ClassList EOF

تعريف العيّلة (Class Definition)

ClassDefinition \rightarrow **كلاس** Identifier { FunctionList }

قائمة الكلاسات (classList)

ClassList → ClassDefinition ClassList | ε

قائمة الدوال (FunctionList)

FunctionList → FunctionStmt FunctionList | ε

ملاحظة مهمة: يجب أن تحتوى قائمة الدوال في إحدى الالسات على دالة واحدة فقط اسمها **سامو عليكم** (الدالة الرئيسية).

تعريف الدوال والمعاملات

تعريف الدالة (FunctionStmt)

FunctionStmt → [جاعد [Type] Identifier (ParameterList) BlockStmt

: حيث

- Identifier هو اسم الدالة
- الدالة الرئيسية: يجب أن تكون دالة واحدة فقط باسم سامو عليكم
- جاعد static تشير إلى

قائمة المعاملات (ParameterList)

ParameterList → [Parameter (, Parameter)*]

Parameter → Type Identifier

 لتبسيط عملية TokenType.Rakam ملاحظة: في الكود البرمجي، تم افتراض أن المعاملات يجب أن تكون من نوع رقم التحليل. يمكن توسيعها لتشمل أنواعاً أخرى لاحقاً.

الكتل والجمل البرمجية ◆

الكتلة البرمجية (BlockStmt)

BlockStmt → { StatementList }

قائمة الجمل (StatementList)

StatementList → Statement StatementList | ε

الجملة (Statement)

Statement → Declaration

| Assignment
| IfStmt
| WhileStmt
| ForStmt
| PrintStmt
| ReturnStmt
| BlockStmt

❖ أنواع الجمل البرمجية

التصرير عن متغير (Declaration)

Declaration → Type Identifier [= Expression] ;

: حيث

Type | رقم | كلام | كسر | صح_غلط → رقم | كلام | كسر | صح_غلط

الإسناد (Assignment)

Assignment → Identifier = Expression ;

جملة الشرط (IfStmt)

IfStmt → لو (Condition) Statement [و لا Statement]

جملة التكرار (WhileStmt)

WhileStmt → علطول (Condition) Statement

جملة التكرار (ForStmt)

ForStmt → لف ([Declaration | Expression] ; [Condition] ; [Expression]) Statement

جملة الطباعة (PrintStmt)

PrintStmt → اكتب Expression ;

جملة الإرجاع (ReturnStmt)

ReturnStmt → الجوف [Expression] ;

❖ التعبيرات (Expressions)

التعبير (Expression)

Expression → Comparison

المقارنة (Comparison)

Comparison → Term (RelOp Term) *

: حيث

RelOp → == | != | > | >= | < | <=

الحد (Term)

Term → Factor (AddOp Factor) *

: حيث

AddOp → + | -

العامل (Factor)

Factor → Postfix (MulOp Postfix) *

: حيث

MulOp → * | /

الزيادة اللاحقة (Postfix)

Postfix → Primary [++]

الأولي (Primary)

Primary → Literal | Identifier | (Expression)

: حيث

Literal → NumberLiteral | StringLiteral

ملاحظات إضافية

الدالة الرئيسية (Main Function)

للبرنامـج، ويجب أن تكون نقطـة الدخـول (Entry Point) لـتمثـيل نقطـة الدخـول (SamoAlikom) يـجب أن يـحتـوي البرـنامـج عـلـى دـالـة واحـدة فـقـط باـسـم سـامـو عـلـيـكـمـ. هـذـه الدـالـة ضـمـن إـحـدى الـكـلاـسـات المـعـرـفـةـ.

● (AST Nodes) هيـكل شـجـرة بنـاء الجـملـة المـجـرـدة

العقدـةـ (AST) لـتمـثـيل البنـية الجديدة متـعدـدة الـكـلاـسـاتـ، يـجب أن تـتـضـمـن شـجـرة بنـاء الجـملـة المـجـرـدةـ:

العقدـةـ	الوصـفـ
ProgramNode	يمـثل البرـنامـج بالـكـاملـ، ويـحتـوي عـلـى قائـمة الـكـلاـسـاتـ
ClassNode	يمـثل تعـرـيف العـيـلةـ الـواـحـدةـ
FunctionNode	يمـثل تعـرـيف الدـالـةـ
VarDeclNode	التـصـرـيـحـ عـنـ متـغـيرـ (Declaration)
AssignmentNode	جملـةـ الإـسـنـادـ
PrintNode	جملـةـ الطـبـاعـةـ (PrintStmt)
IfNode	جملـةـ الشـرـطـ (IfStmt)
WhileNode	جملـةـ التـكـرارـ (WhileStmt)
ForNode	جملـةـ التـكـرارـ (ForStmt)
ReturnNode	جملـةـ الإـرـجـاعـ (ReturnStmt)
BlockNode	الـكـتـلـةـ الـبـرـمـجـيـةـ (BlockStmt)
BinaryOpNode	(الـتـعـبـيرـاتـ الثـانـيـةـ) مـثـلـ الجـمـعـ وـالـمـقـارـنـةـ
LiteralNode	الـقـيـمـ الثـابـتـةـ
VariableNode	المـتـغـيرـاتـ
PostfixIncrementNode	عملـيةـ الـزـيـادـةـ الـلـاحـقةـ

● خـلاـصـةـ

هـذـهـ المـسـتـنـدـ يـحدـدـ القـوـاـعـدـ النـحـوـيـةـ الـكـامـلـةـ لـلـغـةـ الـبـرـمـجـةـ، بـدـءـاـ منـ البرـنـامـجـ الـكـامـلـ وـصـوـلـاـ إـلـىـ أـصـغـرـ التـعـبـيرـاتـ. يـمـكـنـ استـخـدـامـ هـذـهـ القـوـاـعـدـ لـبنـاءـ محلـلـ المـنـاسـبـةـ ASTـ قادرـ عـلـىـ تـحلـيلـ الـكـوـدـ المـكـتـوبـ بـهـذـهـ اللـغـةـ وـبنـاءـ شـجـرةـ (Parser)ـ لـغـوـيـ.