

(Parser Examples) أمثلة المحلل اللغوي

مراجعة البنية النحوية

(Grammar) قواعد النحو

ملاحظة عن التعليقات

هذا يعني أن الـ (Lexer) التعليقات (التي تبدأ بـ // وتستمر حتى نهاية السطر) يتم التعامل معها وتجاهلها بالكامل بواسطة المحلل اللغوي. تمثّل التعليقات، وبالتالي يمكن كتابة التعليقات في أي مكان في الكود دون التأثير على بنية الجملة النحوية (Tokens) لن يتلقى أي رموز Parser.

◆ القواعد النحوية الأساسية

(Program) البرنامج

Program \rightarrow ClassList EOF

(Class Definition) تعريف العيلة

ClassDefinition \rightarrow عيلة Identifier { FunctionList }

(ClassList) قائمة الكلاسات

ClassList \rightarrow ClassDefinition ClassList | ϵ

(FunctionList) قائمة الدوال

FunctionList \rightarrow FunctionStmt FunctionList | ϵ

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

◆ تعريف الدوال والمعاملات

تعريف الدالة (FunctionStmt)

FunctionStmt \rightarrow [حيسبة [جاعد [Type] Identifier (ParameterList) BlockStmt


حيث:

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

قائمة المعاملات (ParameterList)

ParameterList \rightarrow [Parameter (, Parameter)*]

Parameter \rightarrow Type Identifier

لتبسيط عملية (TokenType.Rakam) **ملاحظة:** في الكود البرمجي، تم افتراض أن المعاملات يجب أن تكون من نوع رقم  التحليل. يمكن توسيعها لتشمل أنواعًا أخرى لاحقًا.

◆ الكتل والجمل البرمجية

الكتلة البرمجية (BlockStmt)

BlockStmt \rightarrow { StatementList }

قائمة الجمل (StatementList)

StatementList \rightarrow Statement StatementList | ϵ

الجملة (Statement)

Statement \rightarrow Declaration
| Assignment
| IfStmt
| WhileStmt
| ForStmt
| PrintStmt
| ReturnStmt
| BlockStmt

♦ أنواع الجمل البرمجية

(Declaration) التصريح عن متغير

Declaration \rightarrow Type Identifier [= Expression] ;

حيث:

Type \rightarrow رقم | كلام | كسر | صح_غلط

(Assignment) الإسناد

Assignment \rightarrow Identifier = Expression ;

(IfStmt) جملة الشرط

IfStmt \rightarrow لو (Condition) Statement [وإلا Statement]

(WhileStmt) جملة التكرار

WhileStmt \rightarrow علطول (Condition) Statement

(ForStmt) جملة التكرار

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

(PrintStmt) جملة الطباعة

PrintStmt \rightarrow اكتب Expression ;

(ReturnStmt) جملة الإرجاع

ReturnStmt \rightarrow الجوف [Expression] ;

♦ التعبيرات (Expressions)

(Expression) التعبير

Expression \rightarrow Comparison

المقارنة (Comparison)

Comparison \rightarrow Term (RelOp Term) *

حيث:

RelOp \rightarrow == | != | > | >= | < | <=

الحد (Term)

Term \rightarrow Factor (AddOp Factor) *

حيث:

AddOp \rightarrow + | -

العامل (Factor)

Factor \rightarrow Postfix (MulOp Postfix) *

حيث:

MulOp \rightarrow * | /

الزيادة اللاحقة (Postfix)

Postfix \rightarrow Primary [++]

الأساسي (Primary)

Primary \rightarrow Literal | Identifier | (Expression)

حيث:

Literal \rightarrow NumberLiteral | StringLiteral

ملاحظات إضافية

الدالة الرئيسية (Main Function)

للبرنامج، ويجب أن تكون (Entry Point) لتمثل نقطة الدخول (SamoAlikom) يجب أن يحتوي البرنامج على دالة واحدة فقط باسم سامو عليكم هذه الدالة ضمن إحدى الكلاسات المعرفة.

AST Nodes هيكل شجرة بناء الجملة المجردة

العقد التالية (AST) لتمثيل البنية الجديدة متعددة الكلاسات، يجب أن تتضمن شجرة بناء الجملة المجردة:

الوصف	العقدة
يمثل البرنامج بالكامل، ويحتوي على قائمة الكلاسات	ProgramNode
يمثل تعريف العيلة الواحدة	ClassNode
يمثل تعريف الدالة	FunctionNode
(Declaration) التصريح عن متغير	VarDeclNode
جملة الإسناد	AssignmentNode
جملة الطباعة (PrintStmt)	PrintNode
جملة الشرط (IfStmt)	IfNode
جملة التكرار (WhileStmt)	WhileNode
جملة التكرار (ForStmt)	ForNode
جملة الإرجاع (ReturnStmt)	ReturnNode
الكتلة البرمجية (BlockStmt)	BlockNode
(التعبيرات الثنائية) مثل الجمع والمقارنة	BinaryOpNode
القيم الثابتة	LiteralNode
المتغيرات	VariableNode
عملية الزيادة اللاحقة	PostfixIncrementNode

خلاصة

هذا المستند يحدد القواعد النحوية الكاملة للغة البرمجة، بدءاً من البرنامج الكامل وصولاً إلى أصغر التعبيرات. يمكن استخدام هذه القواعد لبناء محلل المناسبة AST قادر على تحليل الكود المكتوب بهذه اللغة وبناء شجرة (Parser) لغوي.