

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA

Curso: Organizacion De Lenguajes Y Compiladores 2

Sección: B

Catedrática: Damihan Antoni Morales Cuyuch

Manual

Nombre: Gonzalo Fernando Pérez Cazún

Carnet: 202211515

Fecha: 17/09/2024

## DESCRIPCION

Se requiere el desarrollo de un intérprete para el lenguaje de programación OakLand. Este lenguaje, aunque inspirado en la sintaxis de Java, no se define por esta característica principal. OakLand destaca por su capacidad para manejar múltiples paradigmas de programación, incluyendo la orientación a objetos, la programación funcional y la procedimental. Además, se debe desarrollar una plataforma sencilla pero suficientemente robusta para permitir la creación, apertura, edición e interpretación del código en OakLand.

La implementación del intérprete debe considerar la integración de estos paradigmas de programación y proporcionar una interfaz que facilite la manipulación del código. La plataforma deberá incluir funcionalidades para editar y ejecutar programas en OakLand, garantizando la compatibilidad y la eficiencia en la interpretación del lenguaje.

## GRAMATICA

<programa> ::= <dcl>\*

<Declaracion> ::= <DeclaracionVar>

| <FuncDcl>

| <Stmt>

<DeclaracionVar> ::= <Tipo> <Identificador> [ "=" <Expresion> ] ";"

<FuncDcl> ::= <TipoFuc> <Identificador> "(" [ <Parametros> ] ")" <Bloque>

<Parametros> ::= <Parametro> [ ",", <Parametros> ]

<Parametro> ::= <TipoFuc> <Identificador>

<Stmt> ::= "System.out.println" "(" <ExpresionPrintln> ")" ";"

| <Bloque>

| "if" "(" <Expresion> ")" <Stmt> [ "else" <Stmt> ]

| "switch" "(" <Expresion> ")" "{" <Casos>\* [ "default" ":" <BloqueAux> ] "}"

| "while" "(" <Expresion> ")" <Stmt>

| "for" "(" <ForInit> <Expresion> ";" <Expresion> ")" <Stmt>

| "break" ";"

| "continue" ";"

| "return" [ <Expresion> ] ";"

| <Expresion> ";"

<Bloque> ::= "{" <Declaracion>\* "}"

<Casos> ::= "case" <Expresion> ":" <BloqueAux>

<BloqueAux> ::= <Declaracion>\*

<ForInit> ::= <DeclaracionVar>

| <Expresion> ";"

| " ; "

<ExpresionPrintln> ::= <Expresion> [ " , " <ExpresionPrintln> ]

| <Expresion>

<Expresion> ::= <Ternario>

<Ternario> ::= <Asignacion> "?" <Asignacion> ":" <Asignacion>

| <Asignacion>

<Asignacion> ::= <Identificador> "=" <Asignacion>

| <AsignacionOp>

<AsignacionOp> ::= <Identificador> "+=" <Asignacion>

| <Identificador> "-=" <Asignacion>

| <Identificador> "++" [ <Expresion> ]

| <Identificador> "--" [ <Expresion> ]

| <Or>

<Or> ::= <And> [ "||" <And> ]\*

<And> ::= <Comparacion> [ "&" <Comparacion> ]\*

<Comparacion> ::= <Relacional> [ ("==" | "!=") <Relacional> ]\*

<Relacional> ::= <Suma> [ ("<" | ">" | "<=" | ">=") <Suma> ]\*

<Suma> ::= <Multiplicacion> [ ("+" | "-") <Multiplicacion> ]\*

<Multiplicacion> ::= <Unaria> [ ("\*" | "/" | "%") <Unaria> ]\*

<Unaria> ::= ("-" | "!") <Unaria>

| <Nativo>

| <Llamada>

| <LlamadaEs>

<Llamada> ::= <Nativo> "(" [ <Argumentos> ] ")"

<Argumentos> ::= <Expresion> [ "," <Argumentos> ]

<LlamadaEs> ::= <Identificador> <Expresion>\*

<idtypeof> ::= "typeof"

<Nativo> ::= [0-9]+ "." [0-9]+

| [0-9]+

| ("true" | "false")

| "'" <chars> "'"

| "" <chars> ""  
| "(" <Expresion> ")"  
| <Identificador>

<chars> ::= <character>\*  
| <escapeSequence>

<escapeSequence> ::= "\\"  
| "\\\"  
| "\\n"  
| "\\r"  
| "\\t"

<Tipo> ::= "int"  
| "float"  
| "string"  
| "boolean"  
| "char"  
| "var"

<TipoFuc> ::= "int"  
| "float"  
| "string"  
| "boolean"  
| "char"  
| "void"  
| <Identificador>

<Identificador> ::= [a-zA-Z] [a-zA-Z0-9]\*

<Comentarios> ::= "//" [^\n]\*

| "/"\* [^("\*/")] "\*" "/"