

Informe

Tema: Analizador Léxico y Sintáctico

Introducción

La principal razón de ser del lenguaje de programación propuesto es dar a conocer a los estudiantes y profesionales de nuestra rama cómo se vería un lenguaje de programación cuyas principales palabras reservadas sean caracteres del chino mandarín.

Descripción

El lenguaje se llama ChinitoScript, así es, script con la letra l. El lenguaje ChinitoScript admite una función main '主', funciones externas y dentro de ambas puede haber bucles, condicionales, sentencias y expresiones. La sintaxis de este lenguaje es muy similar a la de C++ con algunas características incluidas de Python. Así sería un 'Hola mundo' en ChinitoScript.

```
//Hola mundo en ChinitoScript
主 {
    打印 ("Hola mundo!");
}
```

Especificación Léxica

La especificación léxica de un compilador es una parte crucial del proceso de análisis léxico, que es la primera fase de la compilación. Durante esta etapa, el compilador lee el código fuente y lo descompone en componentes llamados tokens reconocidos a través de expresiones regulares. Los tokens son las unidades básicas del lenguaje, similares a las palabras y símbolos en un idioma natural. Las expresiones regulares son un patrón de búsqueda para el reconocimiento de tokens en la entrada del código fuente.

Tokens	Expresión Regulares
MLCOMM	$/ * [^"] * \setminus * /$
ILCOMM	$(//.*)$
MAIN	主
RETURN	回
FOR	圈
IF	如果
ELSE	否则
PRINT	打印
DTINTEGER	整数
DTFLOAT	小数
DTSTRING	文本
DTCHAR	字
DTBOOL	逻辑
DTVOID	空
IDENTIFIER	$([a - z][A - Z])([a - z][A - Z][0 - 9])^*$
EQUAL	$==$
ASSIGN	$=$
VINTEGER	$[0 - 9]^+$
VFLOAT	$[0 - 9]^+ [0 - 9]^+$
VSTRING	$/ * [^"] * \setminus * /$
VCHAR	$"[^"] * "$
VTRUE	对
VFALSE	错
OPINC	$++$
PLUS	$+$
OPDC	$--$
MINUS	$-$
MULT	$*$
DIV	$/$
MOD	$\%$
AND	$\&\&$

Tokens	Expresión Regulares
OR	
NOT	!
GREATERTH	>
LESSTH	<
GREATEREQTH	>=
LESSEQTH	<=
NOTEQUAL	!=
OKEY	{
CKEY	}
OPAR	(
CPAR)
DOTCOMMA	;
COMMA	,

Gramática

```

PROGRAM -> MAIN '
PROGRAM -> FUNCTION '
FUNCTION' -> DATATYPE IDENTIFIER OPAR PARAMETERS CPAR OKEY
        STATEMENTS LOOPS IFELSES CKEY
PARAMETERS -> PARAMETER MORE_PARAMETERS
MORE_PARAMETERS -> COMMA PARAMETER MORE_PARAMETERS
MORE_PARAMETERS -> ''
PARAMETER -> DATATYPE IDENTIFIER
MAIN' -> MAIN OKEY STATEMENTS LOOPS IFELSES CKEY
MAIN' -> ''
LOOPS -> FOR OPAR ASSIGNMENT CONDITION UPDATE CPAR LOOPS
        OKEY STATEMENTS LOOPS IFELSES CKEY
CONDITION -> EXPRESSION DOTCOMMA
UPDATE -> INCREMENT
INCREMENT -> IDENTIFIER OPUPDATE
OPUPDATE -> OPINC
OPUPDATE -> OPDC
LOOPS -> ''

```

```

IFELSES -> IF OPAR EXPRESSION CPAR OKEY STATEMENTS LOOPS
    IFELSES CKEY ELSE '
IFELSES -> ''
ELSE' -> ELSE OKEY STATEMENTS LOOPS IFELSES CKEY
ELSE' -> ''
STATEMENTS -> PRINT OPAR EXPRESSION CPAR DOTCOMMA
    STATEMENTS
STATEMENTS -> ASSIGNMENT STATEMENTS
STATEMENTS -> RETURN EXPRESSION DOTCOMMA STATEMENTS
STATEMENTS -> ''
ASSIGNMENT -> DATATYPE IDENTIFIER ASSIGN EXPRESSION
    DOTCOMMA
EXPRESSION -> TERM EXPRESSION '
EXPRESSION' -> PLUS TERM EXPRESSION '
EXPRESSION' -> MINUS TERM EXPRESSION '
EXPRESSION' -> AND TERM EXPRESSION '
EXPRESSION' -> OR TERM EXPRESSION '
EXPRESSION' -> EQUAL TERM EXPRESSION '
EXPRESSION' -> NOTEQUAL TERM EXPRESSION '
EXPRESSION' -> GREATERTH TERM EXPRESSION '
EXPRESSION' -> LESSTH TERM EXPRESSION '
EXPRESSION' -> GREATEQTH TERM EXPRESSION '
EXPRESSION' -> LESSEQTH TERM EXPRESSION '
EXPRESSION' -> ''
TERM -> FACTOR TERM '
TERM' -> MULT FACTOR TERM '
TERM' -> DIV FACTOR TERM '
TERM' -> MOD FACTOR TERM '
TERM' -> ''
FACTOR -> DVORID
FACTOR -> OPAR EXPRESSION CPAR
FACTOR -> NOT FACTOR
DVORID -> VINTEGER
DVORID -> VFLOAT
DVORID -> VSTRING
DVORID -> VCHAR
DVORID -> VTRUE
DVORID -> VFALSE
DVORID -> IDENTIFIER

```

```
DATATYPE -> DTINTEGER  
DATATYPE -> DTFLOAT  
DATATYPE -> DTSTRING  
DATATYPE -> DTCHAR  
DATATYPE -> DTBOOL  
DATATYPE -> DTVOID
```

Implementación

Revisar [repositorio](#) que contiene

- Analizador Léxico
- Analizador Sintáctico
- Gramática
- Tabla Sintáctica
- Parser LL1
- Abstract Syntax Tree