DOCUMENTACIÓN DEL LEGUAJE

Rodrigo Ortiz Gonzalez

Lenguajes y Autómatas II

Contenido

Documentación del Lenguaje

Este lenguaje de programación está diseñado para ser intuitivo y fácil de usar, con el propósito de facilitar el aprendizaje y fomentar la creatividad. Ofrece estructuras básicas como control de flujo, expresiones, funciones y asignaciones, permitiendo a los usuarios escribir código claro, legible y bien organizado.

Además, está especialmente pensado para ser accesible a niños y principiantes en programación, proporcionando una experiencia educativa amigable y motivadora. Su diseño intuitivo y su enfoque pedagógico buscan inspirar a los nuevos programadores mientras adquieren habilidades fundamentales.

El lenguaje está desarrollado con ANTLR4 para definir su gramática, y su compilador se implementa en C# utilizando Visual Studio 2022. Esto asegura una base sólida, flexible y moderna para procesar el código.

Gramática General

Estructura de un programa

Un programa en este lenguaje está compuesto por una serie de líneas (“line”) que pueden ser declaraciones, bloques condicionales, o bucles. El programa finaliza con `EOF` (siglas de End Of File o Fin Del Archivo).

antlr

grammar Simple;

program: line* EOF;

Declaraciones permitidas

Declaración o ejecución de funciones

Sintaxis:`IDENTIFIER '(' [expression (',' expression)*] ')' ';'`

Ejemplo:

escribir("Hola Mundo");

Asignación

Sintaxis: `IDENTIFIER 'asigno' expression;`

Ejemplo:

x asigno 5;

Bloques condicionales

Condicionales (`si` y `sino`)

Sintaxis:

```
    si (expression) {
//Codigo
    } sino {
//Codigo
    }
```

Ejemplo:

```
si x > 0 {
   escribir("El número es positivo");
}
sino {
   escribir("El número no es positivo");
}
```

Bucles

Bucle (`bucle` o `hasta`)

Sintaxis:

```
    bucle (expression) {
       // Bloque de código
    }
```

Ejemplo:

```
bucle x < 5 {
   x asigno x + 1;
   escribir("El valor de x es: " + x+);
}
```

Operadores

Operadores Aritméticos	
Multiplicación	*
División	/
Suma	+
Resta	-

Ejemplo:

y asigno 3 + 4;

Operadores de Comparación	
Igualdad	==
Diferencia	!=
Menor que	<
Mayor que	>
Menor o igual	<=
Mayor o igual	>=

Ejemplo:

```
si (x == y) {
   escribir("Son iguales");
}
```

Operadores Booleanos	
AND	and
OR	or
XOR	xor

Ejemplo:

```
si (a and b) {
   escribir("Ambos verdaderos");
}
```

Operadores Avanzados	
Exponenciación	estrella
Raíz cuadrada	solecito

Ejemplo:

x asigno 3 estrella 2;

resultado asigno x solecito x;

Constantes y Tipos de Datos		
Números enteros	INTEGER: [0-9]+	**Ejemplo:** `123`
Números flotantes	FLOAT: [0-9]+ '.' [0-9]*	**Ejemplo:** `3.14`
Cadenas de texto	STRING: ('"' ~'"'* '"') | ('\'' ~'\''* '\'')	**Ejemplo:** `'Hola'`, `"Mundo"`
Booleanos	BOOL: 'verdadero' | 'falso'	**Ejemplo:** `verdadero`, `falso`
Nulo	NULL: 'null'	**Ejemplo:** `null`

Bloques de Código

Los bloques de código se definen entre llaves `{}` y pueden contener varias líneas.

Ejemplo:

```
{
   x asigno 10;
   escribir(x);
}
```

Espacios en blanco

El lenguaje ignora los espacios en blanco y saltos de línea.

antlr

```
WS: [ \t\r\n]+ -> skip;
```

Identificadores

Los identificadores pueden ser letras, números y guiones bajos, pero no pueden comenzar con un número.

antlr

```
IDENTIFIER: [a-zA-Z_][a-zA-Z0-9_]*;
```

Cierre de sentencias

En este lenguaje, todas las sentencias deben cerrarse con un punto y coma (;).

Ejemplo de una sentencia válida:

```
miVariable = 10;
```

Implementación

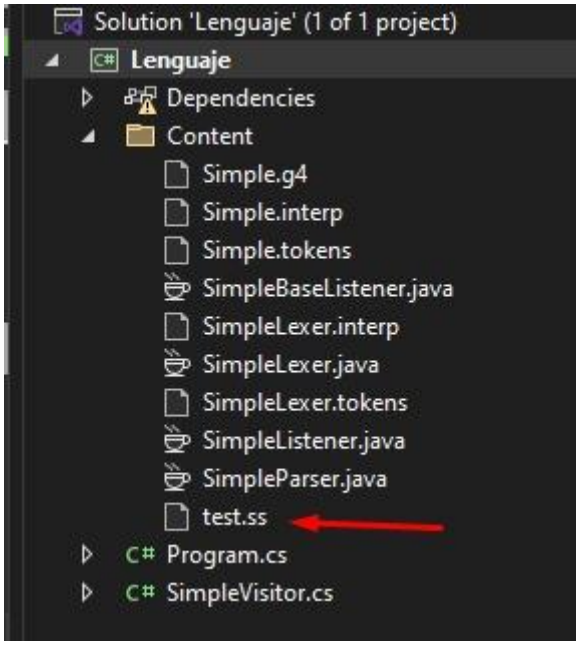
El lenguaje soporta una implementación que permite manejar variables predefinidas, como `PI` y `E`, y funciones predefinidas como `escribir` para imprimir mensajes:

Variables Predefinidas	
PI	Representa el valor de pi
E	Representa el valor de la constante de Euler

Funciones Predefinidas		
escribir	Imprime un mensaje en la salida	**Ejemplo**: escribir("Hola Mundo");

Como usarlo

Para ejecutar el compilador en Visual Studio Code 2022, primero asegúrate de tener configurado el entorno con C# y .NET; el archivo **test.ss**, el cual se encuentra el directorio **Content** dentro de la carpeta con el programa, la cual actúa como la entrada para el compilador, por lo que cualquier código que desees compilar debe escribirse o modificarse dentro de este archivo antes de ejecutar el programa.



Para compilar el código escrito, simplemente haz clic en el botón "Correr" en Visual Studio 2022. Al iniciar, los resultados de la compilación se mostrarán en una ventana del símbolo del sistema, ya que el proyecto está configurado como una aplicación de consola en C#.

Ejemplo Completo

```
simple

x asigno 0;

p asigno 0;

p asigno p + 5;

escribir(p);

p asigno p - 1;

escribir(p);

p asigno p * 2;

escribir(p);

p asigno p / 2;

escribir(p);

p asigno p estrella 2;

escribir(p);

p asigno p solecito p;

escribir(p);

bucle x < 5 {
   x asigno x + 1;
   escribir("El valor de x es: " + x+);
}

si x > 0 {
   escribir("El número es positivo");
}

sino {
   escribir("El número no es positivo");
}
```