OLC-1C

PROYECTO2
MANUAL TECNICO

Selvin Orlando Hernández
201700603

Ing. Kevin Lajpop

Auxliar. Sandra Jimenenz

Descripción de la solución:

Para este proyecto se implementó la arquitectura Cliente-Servidor, de tal forma que los servicios que ofrece la api permiten al cliente presentar la funcionalidad de un intérprete para un lenguaje personalizado llamado CS.
Cuenta con un entorno de trabajo, cuya finalidad será proporcionar ciertas herramientas que serán de utilidad para el usuario, la función principal será el ingreso de código fuente el cual será analizado.

Se aplicaron los conocimientos de la fase de análisis léxico y sintáctico de un compilador para la realización de un intérprete sencillo, con las funcionalidades principales como diversas librerías, métodos y clases para crear la solución óptima esperada, dichas funciones se especifican más adelante del manual.

Clases Implementadas

A continuación, se detallan algunas de las clases principales utilizadas en el proyecto, puesto que el cliente se realizó mediante html y js estos no poseen clases como tal.

Clases del Servidor

- Ast: Esta se encarga de inicializar los métodos ejecutar de las clases que implementan las interfaces de instrucción.
- Errores: esta clase es la que nos permite crear objetos de tipo error ya sean léxicos, sintácticos y semánticos que posteriormente estaremos mostrando en los reportes de errores.
- Nodo: esta clase nos permite generar los nodos del AST que posteriormente se graficaran
- Aritmética: Contiene las operaciones validas dentro de la aplicación, sumas, restas, multiplicaciones, divisiones, potencias, modulo y negación unaria. El constructor utiliza el mismo que la clase Operación.
- Logica: Contiene los operadores lógicos, and, or y not. El constructor utiliza el mismo que la clase Operación.
- Operacion: Clase para el manejo de operaciones del programa
- Relacionales: Instrucciones para las operaciones relacionales que son aceptadas por la gramática.
- Casteo: Es una forma de indicar al lenguaje que convierta un tipo de dato en otro, por lo que, si queremos cambiar un valor a otro tipo.
- FNativas: Funciones nativas del programa como tolower, toupper, entre otras.
- Identificador: Obtiene y guarda el identificador de una función, variable o metodo.

- Primitivos: Guarda los valores primitivos los cuales son: enteros, dobles, cadenas, booleanos, caracteres.
- Ternario: Funciona para dicha instrucción la cual es un operador que hace uso de 3 operandos para simplificar la instrucción ‘if’ por lo que a menudo este operador se le considera como un atajo para la instrucción ‘if
- Function: Reconoce las funciones o métodos que la gramática puede aceptar.
- Llamada: Clase para llamar y ejecutar las funciones o métodos reconocidos.
- sentreturn: Retorna el valor de una función, finaliza la ejecución de un método o función y puede especificar un valor para ser devuelto a quien llama a la función.
- Asignacion: Como su nombre lo indica, asigna los diversos valores que puede tener un identificador.
- Caso: Utilizado por la sentencia de control “Switch”, son los casos que la sentencia puede tener.
- Declaracion: Clase para declarar identificadores, verifica el tipo y el identificador el cual debe ser único.
- Sentcontinue: Clase para la instrucción continue la cual puede detener la ejecución de la iteración actual y saltar a la siguiente
- SentBreak: Clase para la instrucción break la cual puede hacer que se salga del ciclo inmediatamente, es decir que el código que se encuentre después del break en la misma iteración no se ejecutara y este se saldrá del ciclo.
- Sentfor: Clase que nos permite ejecutar N cantidad de veces la secuencia de instrucciones que se encuentra dentro de ella la cual se comporta igual que el for de cualquier otro lenguaje de programación.
- SentIf: Clase que nos permite ejecutar las instrucciones sólo si se cumple una condición. Si la condición es falsa, se omiten las sentencias dentro de la sentencia, esto hace que se comporte como un if de cualquier otro lenguaje de programación.
- SentSwitch: Estructura que nos permite crear una sentencia de control Sentwhile: Clase que ejecuta una secuencia de instrucciones mientras la condición de ejecución se mantenga verdadera, se comporta igual que un.
- StartWith: Clase que sirve para ejecutar todo el código generado dentro del lenguaje, se utilizará la sentencia START WITH para indicar que método o función es la que iniciará con la lógica del programa.
- Writeline: Instrucción para imprimir valores en la consola de la aplicación.
- Instruccion: Ejecuta las instrucciones del programa.
- Símbolo: Define los símbolos del lenguaje los cuales son variables, métodos o funciones.
- TablaSimbolos: Guarda las variables, métodos y funciones.
- Tipo: Permite enumerar los tipos del lenguaje, llevar el control de los diversos tipos que la gramática reconoce.
- Controlador: Clase que nos permite llevar el control de errores y la consola de todo el programa.

Requisitos mínimos para el entorno de desarrollo

1	Visual Studio Code 1.61 o superior
2	Express
3	typescript
4	nodemon
5	nodejs 14 o superior
6	Ram 4GB o superior
7	Windows 10/Linux
8	Navegador Ophera, Edge, Firefox,Chrome
9	Espacio en disco 1GB o superior