UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
SEGUNDO SEMESTRE 202

C

MANUAL TÉCNICO
PROYECTO 1

Nombre: Angel Francisco Sique Santos
Carné: 202012039

INTRODUCCIÓN

Un cliente ha solicitado a usted un Pseudo-Parser para que el nuevo personal que no conoce los lenguajes de Python y Golang, pueda aplicar sus conocimientos en pseudocódigo y utilizando una aplicación, traducir este código y ver cómo se comportan las diferentes sintaxis de cada uno de los lenguajes ya que para cada uno existen diferentes características, por lo cual, se le solicita a usted que a partir de sus conocimientos en compiladores haga una implementación de las primeras 2 fases de un compilador y ejecute una traducción con la entrada de pseudocódigo a Python y Golang. Además que se pueda visualizar el diagrama de flujo resultante de la entrada, para esto debe utilizar las herramientas de JFLEX y CUP e implementar su solución en el lenguaje de programación JAVA.

REQUISITOS MÍNIMOS

500mb de disco duro

2gb de RAM

Windows 7

Equipo Intel Pentium o superior

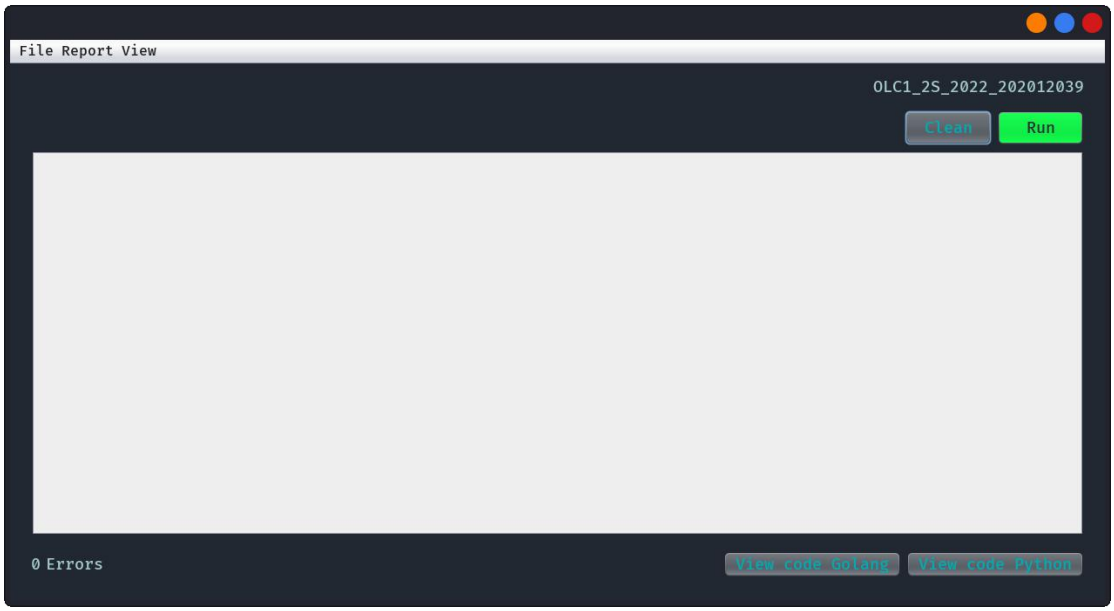
Herramientas de análisis léxico y sintáctico: JFlex/CUP

Java 14

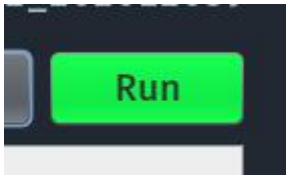
Graphviz

OPCIONES DEL PROGRAMA

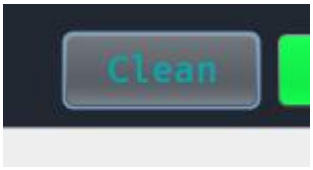
1. En esta parte creamos la interfaz gráfica, en este caso se hizo uso de la herramienta de drag and drop incuida en netbeans para realizarla.



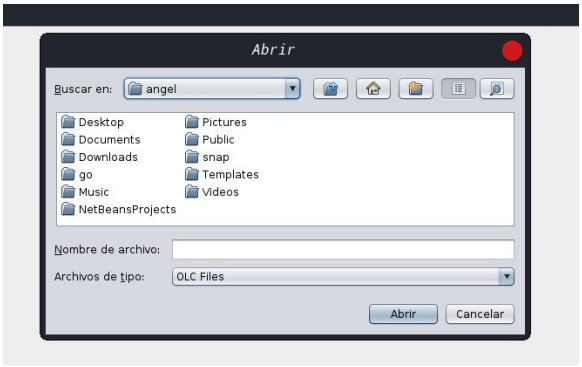
2. En el botón Run leemos el archivo de entrada, creamos el AST además del diagrama de flujo del código ingresado y también creamos un HTML con los errores sintáctocos y léxicos encontrados.



3. El botón clean limpia todo el cuadro de texto.

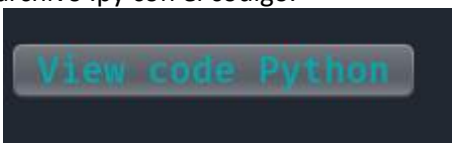


4. La opción Open File abre un espacio para escoger un archivo de entrada e ingresarlo al cuadro de texto.

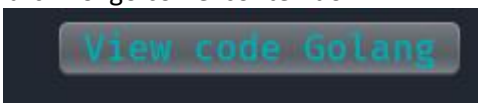


5. La opción Save File guarda el contenido del cuadro de texto en un archivo con el nombre que el usuario elija.

6. La opción View Python Code traduce el código ingresado a código Python y lo muestra en el cuadro de texto. Además de que crea el HTML con los errores encontrados, tanto léxicos como sintácticos y crea un archivo .py con el código.



7. La opción View Golang Code traduce el código ingresado a código Golangos, tanto léxicos como sintácticos y crea un archivo .go con el contenido.



8. La opción Errors abre el HTML generado con la lista de todos los erroes léxicos y sintácticos.

Linea	Columna	Descripcion	Tipo
11	53	No se esperaba este componente: //ERROR sintactico, reportar que falta un punto y coma, recuperarse del error	Sitactico
56	70	No se esperaba este componente: es	Sitactico
58	34	No se esperaba este componente: c	Sitactico
66	31	No se esperaba este componente: "Organizacion de lenguajes y compiladores 1"	Sitactico
107	0	No se esperaba este componente: if	Sitactico
109	24	No se esperaba este componente: Numeror	Sitactico
116	0	No se esperaba este componente: fin_si	Sitactico
137	15	No se esperaba este componente: 1	Sitactico
139	27	No se esperaba este componente: 3	Sitactico
140	35	No se esperaba este componente: ,	Sitactico
7	0	Componente @ no reconocido.	Lexico
7	1	Componente $ no reconocido.	Lexico
7	2	Componente & no reconocido.	Lexico
56	72	Componente _ no reconocido.	Lexico
60	0	Componente @ no reconocido.	Lexico
69	0	Componente $ no reconocido.	Lexico
109	54	Componente % no reconocido.	Lexico
137	16	Componente _ no reconocido.	Lexico
141	9	Componente _ no reconocido.	Lexico