

Instituto Tecnológico de Costa Rica

Primer Avance sistema Compilador

Integrantes:

Josué Arce González

Daniel Montero Carvajal

Curso de Compiladores e Interpretes

25/03/2018

Contenido

Análisis del lenguaje.....	3
Que puede hacer.....	3
Que no puede hacer	3
Soluciones e implementaciones	4
.....	5
Resultados obtenidos.....	6
Conclusiones.....	7
Bibliografías	7

Análisis del lenguaje

Que puede hacer

- Asignar variables por medio del comando **let**.
- Crear funciones con 1 o más parámetros.
- Asignar variables de tipo **string**, **integer** y **booleanos**.
- Asignar a una variable el resultado de una función, aunque no retorna nada.
- Asignar a una variable el resultado de una operación matemática. Uno de los valores de la operación puede ser otra variable.
- Crear listas y diccionarios.
- Crear una lista que contenga números, strings, booleanos, listas, otras variables y funciones.
- Utilizar el condicional **if** y **else**.
- No incluir un punto y coma al final de una asignación o función.
- Crear 2 variables con el mismo nombre.
- Crear funciones dentro de otras funciones.

Que no puede hacer

- Crear funciones sin parámetros.
- Pasar como parámetro a una función algo que no sea una variable
- Asignar una variable dentro de una lista.

Soluciones e implementaciones

Explicación general: el primer avance se encuentra segmentado en 2 partes, la parte visual, el cual es conformada por el editor de texto creado para poder insertar texto y así compilarlo, por otro lado, se encuentra el escáner y el analizador, encargados de recibir el texto insertado, analizarlo y generar una respuesta respecto al mismo, a continuación, se realizará una explicación detallada de cada una de sus partes.

1. Scanner:

- a. Scanner.g4: Encargado de almacenar los identificadores para cada conjunto de caracteres posibles a insertar, en el mismo se puede mencionar los símbolos y palabras reservadas que se asignan a una variable para así poder manejar estas de una mejor manera en el archivo Scanner.java y en otros que más adelante se van a explicar mejor.
- b. Scanner.java: Es el archivo que se produce al generar el archivo anterior (Scanner.g4), en este mismo se pueden encontrar las mismas expresiones que existen en el archivo Scanner.g4 pero traducidos al lenguaje Java, con el sentido de poder manipularlas de una mejor manera, estas expresiones y métodos son utilizados en otros archivos, por lo que la elaboración de estos debió de realizarse de una forma muy cautelosa para que el compilador funcione al 100%.

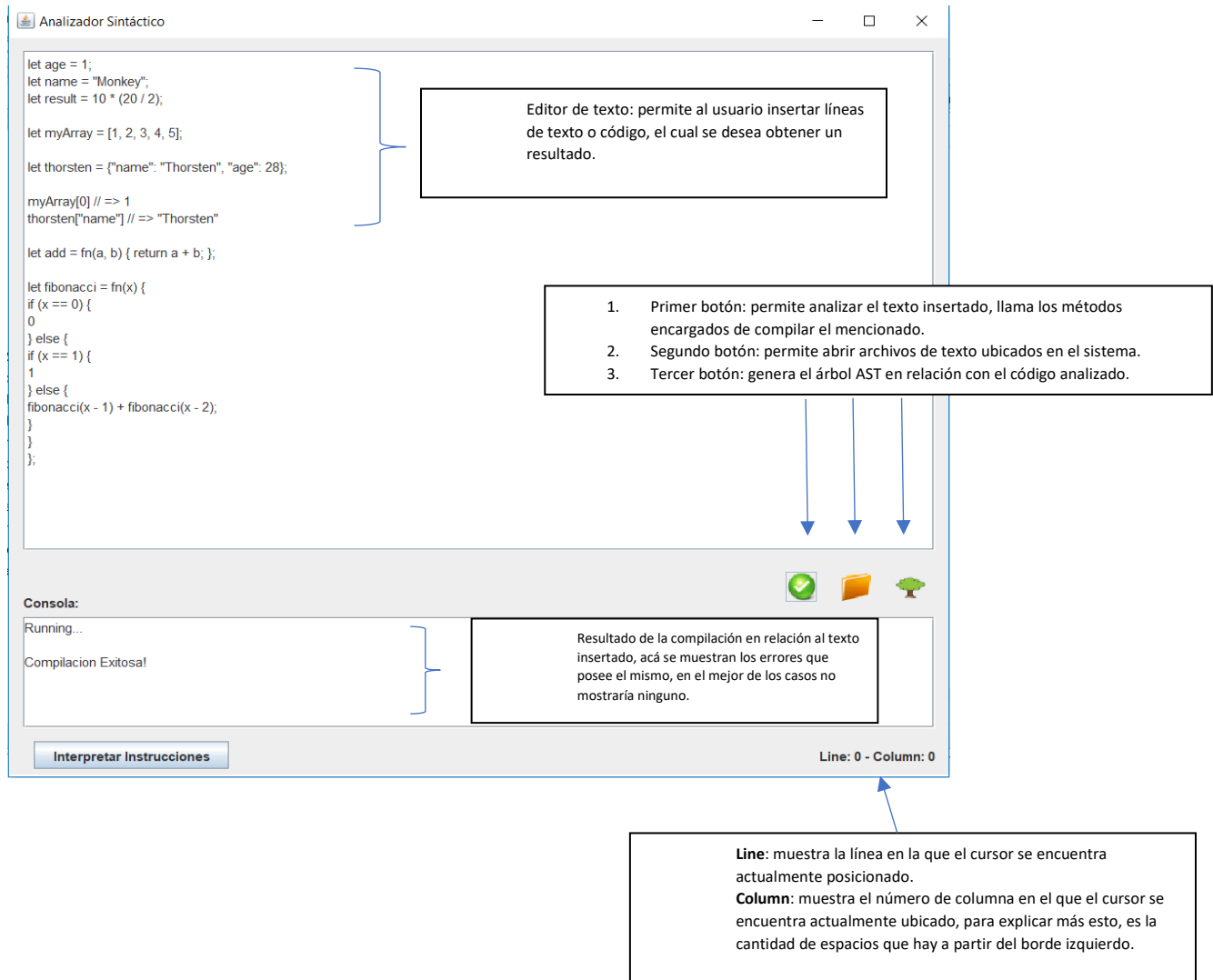
2. Parser:

- a. Parser.g4: Posee todas las reglas que definen la base del compilador, es decir, en este archivo están escritas todas aquellas reglas sintácticas de las cuáles el proyecto va utilizar para poder así valorar y definir que se puede o que no se puede insertar; este archivo es la base, una vez que se terminó de elaborar este archivo se generó el archivo Parser.java.
- b. Parser.java: Resultado de la generación del archivo Parser.g4, este código de extensión java posee las reglas de inserción, pero en el lenguaje Java, este archivo es el que el proyecto va utilizar para condicionar que se puede o no realizar; es la traducción exacta del archivo Parser.g4. Los métodos, variables, entre otros que se encuentran en este se van a utilizar en otros archivos.

3. Editor de texto:

- a. Ventana.class: archivo encargado de darle funcionalidad a la ventana principal, todos los botones poseen una funcionalidad, este archivo se encuentra enlazado a los archivos mencionados anteriormente, tales como el Scanner y el Parser, los cuales van a recibir lo insertado en la ventana principal.

b. Ventana principal:



Resultados obtenidos

Aspecto a Evaluar	Estado (0%-100%)	Justificación en caso de no estar el aspecto al 100%
Scanner	Se encuentra funcionando en perfectas condiciones	N/A
Parser	Se encuentra funcionando en perfectas condiciones	N/A
Funciones ventana principal	Se encuentra funcionando en perfectas condiciones	N/A
Editor de texto(visual)	Entendible para el usuario, se hicieron todas las pruebas posibles, se encuentra en perfectas condiciones.	N/A

Conclusiones

En conclusión, a los resultados obtenidos en este primer avance se puede decir con plena seguridad de que el sistema compilador es capaz de leer un texto insertado, analizarlo y formular una respuesta; para explicar mejor este punto se hicieron pruebas en donde el compilador tuvo la capacidad de entender los errores escritos en el editor y mostrarlos en consola, asimismo, se introdujeron pruebas, las cuales eran correctas y el resultado de estas mostró una compilación sin errores, por lo que se puede decir que el sistema compilador cumple con los requisitos solicitados.

Otro punto el cual es necesario de mencionar es la creación del árbol AST, el cual tiene como función desglosar las instrucciones que obtiene, y así generar una serie de ramas, con distintos procesos, las cuales conllevan a un resultado final, esto es de suma importancia porque nos permite apreciar la forma en que el compilador está leyendo el texto insertado; este mismo funciona de manera correcta, despliega todas las ramas de manera exitosa, los procesos y etapas cubiertas por el compilador.

Bibliografías

Joshi, J. (11 de 02 de 2018). *StackOverflow*. Obtenido de https://stackoverflow.com/questions/5139995/java-column-number-and-line-number-of-cursors-current-position?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa