

Mediante este informe, le voy a llevar a través de todo el proceso de funcionamiento de mi proyecto HULK:

HULK es un intérprete de código simple, que recibe una línea de código y devuelve un valor en caso de no ser una declaración de función (Leer el "ReadMe.md" para más información sobre cómo debería ser el código).

Luego de recibir una línea de código mi proyecto procede de la siguiente forma:

La línea de entrada pasa por la clase Lexer la cual separa dicha entrada en una lista de Tokens, para saber el tipo de elementos con los que voy a trabajar a continuación.

Luego esta lista pasa por la clase Parser, para el parseo de esa entrada utilizo un Parser y un SemiParser. El Parser es el encargado de descifrar el tipo de código o la estructura del código que estoy recibiendo y el SemiParser consiste en un Parser descendente recursivo para las expresiones internas de las estructuras detectadas previamente. Ejemplo:

Al declarar una condicional, el Parser se encarga de detectar la estructura de la condicional y luego el SemiParser es el encargado de parsear las expresiones correspondientes al if, al then y al else.

El Parser me devuelve un AST (abstract syntax tree) el cual es una representación escalable de la lista de tokens lo cual hace más fácil el trabajo a la hora de determinar los valores de retorno finales, para lo cual le paso el árbol a la clase interprete que se encarga de visitar los nodos del árbol y devolver los valores de forma recursiva para así al final escribir en consola el valor final de la evaluación.

En medio de todo el procesamiento, hay métodos para la detección de errores, para en caso de que haya errores en el código que introduzca el usuario, no devolver un resultado erróneo.

Esto ha sido un resumen del funcionamiento de mi proyecto, muchas gracias por su atención.