TRABAJO PRACTICO 2

Detección errores

Materia: Técnicas de Compilación

Profesor: Maximiliano Andrés Eschoyez

Integrantes:

- Albarracin, Juan
- Asmuzi Gali

Problema para resolver

El problema consiste en que nuestro programa sea capaz de poder realizar un control de errores básicos (tanto semánticos como sintácticos).

Herramientas utilizadas

ANTLR4: Herramienta que se usó para construir parsers.

Java 17: Lenguaje el cual contiene el proyecto.

Maven: Controlador y gestor del proyecto.

Visual Studio Code: editor de código elegido junto al plugin de ANTLR4 y Maven.

Solución Planteada

La solución consiste en la implementación de un patrón Singleton, que cuenta con las siguientes clases:

TablaSimbolo: Esta clase representa una tabla de símbolos que almacena información sobre variables y funciones declaradas en el código fuente. Contiene métodos para agregar, buscar y obtener información sobre los símbolos.

Funcion: Esta clase representa una función en el código fuente. Tiene atributos como el tipo de retorno, el nombre de la función y una lista de parámetros. También puede almacenar información adicional, como variables locales y el bloque de código asociado a la función.

Variable: Esta clase representa una variable en el código fuente. Tiene atributos como el tipo de dato, el nombre de la variable y su valor. Puede tener métodos adicionales para realizar operaciones relacionadas con las variables, como la inicialización y la asignación de valores.

ID: Esta clase representa un identificador utilizado en el código fuente, ya sea el nombre de una variable, función u otro elemento. Se utiliza para almacenar información sobre el identificador, como su nombre y su tipo.

CustomError: Esta clase representa un error personalizado que puede ocurrir durante el análisis del código fuente. Puede contener información sobre el tipo de error, la ubicación y otros detalles relevantes. Se utiliza para reportar y manejar errores sintácticos o semánticos en el código fuente.

ErrorListener: Esta clase es un listener personalizado que se utiliza para capturar y manejar errores durante el proceso de análisis. Extiende la clase BaseErrorListener de ANTLR y se conecta al lexer y al parser para interceptar y registrar los errores encontrados.

Estas clases están diseñadas para trabajar juntas y facilitar el análisis del código fuente, la construcción de la tabla de símbolos y la detección de errores sintácticos y semánticos. La TablaSimbolo se utiliza para almacenar información sobre las variables y funciones encontradas durante el análisis, mientras que las clases Funcion, Variable e ID ayudan a estructurar y organizar la información relacionada con las funciones y variables.

El ErrorListener se encarga de capturar y registrar los errores encontrados durante el análisis y se puede utilizar para generar un informe de errores como se mencionó en la consigna del trabajo práctico.

Espero que esta explicación aclare cómo se relacionan las clases y cómo se utilizan en el proyecto. Si tienes alguna pregunta adicional o necesitas más detalles, estaré encantado de ayudarte.

Problemas encontrados para plantear la solución

El principal problema fue que nosotros buscábamos que los errores encontrados en Escucha.java trabajen juntos con lo creado, al final no pudimos lograr implementarlo.

Llegamos a una versión en la que mostraba la cantidad de errores, pero en las descripciones mostraba un solo error, tratamos de cambiar la estructura y no pudimos volver a lo anterior, por eso no pudimos encontrar como volver a implementar todo en App.java

Otro problema que tuvimos también fue de estructura, primero teníamos en mente una clase Error, que se encargue de verificar el tipo de error, a partir de ahí que vaya o a la subclase ErrorSintactico o ErrorSemantico, pero tampoco pudimos dar con la implementación correcta

Conclusión

Se nos dificulto bastante poder realizar la implementación deseada, asimismo podemos decir que el trabajo requiere muchos conocimientos que nos llevo tiempo abordar, ya que, aunque teníamos los conocimientos de la asignatura, no contamos con mucha experiencia en la programación con Java, y creo que fue el principal problema.

Links importantes

Repositorio de GitHub donde se alojó el proyecto ; tener en cuenta que el trabajo practico se encuentra en Branch TP2-TC.

Documentación de ANTLR4

Página oficial de ANTLR4

Como ultima información, recomendamos utilizar alguna distribución Linux para realizar este ejercicio, ya que en Windows debemos cargar previamente los Virtual Enviroments de Java y Maven para que pueda funcionar en VS code.

Por último, también recomendamos usar Java 17, no por algo en especial, sino que esa versión usamos nosotros y no tuvimos problemas al trabajar.