UBP UNIVERSIDAD Blas Pascal

Técnicas de Complicación

2023

Trabajo Práctico Nro 2

Profesor: Eschoyez Maximiliano

Alumnos:

- Mora Colodrero Jonathan
- Gigli Constanza

Introducción

El objetivo de este Trabajo Práctico es aplicar, utilizando ANTRL, los temas sobre Análisis Léxico y Sintáctico desarrollados en clase. El programa por desarrollar tiene como objetivo generar como salida el Árbol Sintáctico (ANTLR) correcto, dado un archivo de entrada en lenguaje java.

Consigna

Dado un archivo de entrada en lenguaje C, se debe generar como salida el Árbol Sintáctico (ANTLR) correcto. Para lograr esto se debe construir un *parser* que tenga como mínimo la implementación de los siguientes puntos:

Errores sintácticos comunes:

- Falta de un punto y coma,
- Falta de apertura de paréntesis,
- Formato incorrecto en lista de declaración de variables

Errores semánticos comunes:

- Doble declaración del mismo identificador, • Uso de un identificador no declarado,
- Uso de un identificador sin inicializar,
- Identificador declarado pero no usado,
- Tipos de datos incompatibles.

Cada error reportado debe indicarse si es sintáctico o semántico.

Desarrollo

Para implementar la Tabla de Símbolos y el Árbol Sintáctico, se desarrolló una serie de clases que tienen roles a lo largo del proceso. Describimos las clases utilizadas:

Tabla de Símbolos: Almacena información relevante sobre los contextos y las variables dentro de cada contexto. Sirve como una estructura de datos que permite organizar y acceder a la información necesaria durante el análisis del código.
ID: Representa un identificador, es una variable dentro del programa. Esta contiene información como el tipo de dato asociado al identificador, su nombre, valor y un indicador para determinar si el identificador ha sido utilizado.
Variable: Es una extensión de la clase ID y se utiliza para representar una variable en particular.
MiListener: Utiliza la Tabla de Símbolos para generar los contextos actuales y mantener un historial de los mismos mientras avanza en el análisis del código fuente y además,se encarga de detectar y reportar diversos errores semánticos.

Conclusión

Esta etapa del proyecto permitió mejorar la capacidad del programa para analizar y reportar errores en el código fuente, lo cual es fundamental para garantizar la calidad y confiabilidad de los programas desarrollados.

Github

Jonathan Mora Colodrero: https://github.com/JoniMora/TC

Constanza Gigli: https://github.com/Mariaubp/Tc