

Universidad Tecnológica De Santiago
UTESA
SISTEMA COORPORATIVO.
FACULTAD ARQUITECTURA E INGENIERIA.
CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES.



INTEGRANTES

-NACHELY RODRIGUEZ VASQUEZ

ASIGNATURA

- COMPILADORES

GRUPO

-INF-920-001

PRESENTADO A

- IVAN MENDOZA

ASUNTO

- TAREA SEMANA 6

FECHA DE ENTREGA

-DOMINGO 12 DE NOVIEMBRE DEL 2023

SANTIAGO DE LOS CABALLEROS
REPUBLICA DOMINICANA

Tarea Semana 6

Informe de Documentación del Traductor de Java a C++

Resumen

Este informe detalla el diseño y desarrollo de un traductor de código fuente que convierte programas escritos en el lenguaje de programación Java a su equivalente en C++. El objetivo principal de este proyecto era permitir a los desarrolladores migrar fácilmente sus aplicaciones Java a C++ sin tener que reescribir todo el código manualmente. El traductor se ha implementado como un compilador capaz de analizar y transformar código fuente Java en código C++ equivalente.

Descripción del Proyecto

Objetivo Principal

El objetivo principal de este proyecto es proporcionar una herramienta eficiente y confiable para traducir programas escritos en Java a C++. Esto se logra a través de un compilador que realiza análisis léxico, sintáctico y semántico del código fuente de Java y genera un código equivalente en C++.

Clave funcional

Análisis Léxico y Sintáctico: El traductor realiza un análisis exhaustivo del código fuente Java para identificar las estructuras léxicas y sintácticas. Se utilizan técnicas de análisis léxico y sintáctico para comprender la estructura del programa.

Mapeo de Funcionalidades: Se realiza un mapeo cuidadoso de las características y funcionalidades específicas de Java a sus equivalentes en C++. Esto incluye la conversión de clases, métodos, variables y otras estructuras fundamentales.

Manejo de Bibliotecas: Se implementa un sistema para manejar las diferencias en las bibliotecas estándar entre Java y C++. Esto garantiza que las funciones y clases utilizadas en el programa original de Java se traduzcan de manera eficiente al equivalente en C++.

Resolución de Dependencias: El traductor gestiona la resolución de dependencias entre clases y bibliotecas, asegurando que la traducción sea coherente y que todas las referencias cruzadas se manejen adecuadamente.

Diseño del Compilador

1. Fases del Compilador:

- Análisis Léxico: Convierte el código fuente Java en tokens.
- Análisis Sintáctico: Construye un árbol sintáctico para representar la estructura del programa.
- Análisis Semántico: Realiza verificaciones semánticas y construye una tabla de símbolos.
- Generación de Código: Traduce el árbol sintáctico a código C++ equivalente.

2. Manejo de Errores:

- Se implementa un sistema robusto de manejo de errores que informa al usuario sobre cualquier problema encontrado durante el proceso de traducción.

Ejemplos de uso

Ejemplo de código Java:

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hola, Mundo!");  
    }  
}
```

Código C++ Generado:

```
#include <iostream>  
  
class HelloWorld {  
public:  
    static void main(int argc, char** argv) {  
        std::cout << "Hola, Mundo!" << std::endl;  
    }  
};
```

Conclusiones

El traductor de Java a C++ proporciona una solución efectiva para la migración de código entre estos dos lenguajes de programación. Se ha diseñado con un enfoque en la precisión y la capacidad de gestionar programas Java de tamaño considerable. La documentación aquí proporcionada sirve como referencia completa para los desarrolladores que deseen utilizar esta herramienta y entender su funcionamiento interno.

Futuras Mejoras

Para futuras versiones, se sugiere explorar:

- Mejoras en el rendimiento del compilador.
- Adición de compatibilidad con características avanzadas de Java y C++.
- Incorporación de opciones de configuración para adaptarse a diferentes estilos de codificación.