

Manual de Usuario - DataForge

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización de Lenguajes y Compiladores 1

Primer Semestre 2024

Catedrático:

- Ing. Kevin Lajpop

Tutor académico (aux):

- Carlos Acabal

Tabla de Contenido

1. [Objetivo General](#)
2. [Objetivos Específicos](#)
3. [Descripción General](#)
4. [Entorno de Trabajo](#)
 - [Editor](#)
 - [Funcionalidades](#)
 - [Características](#)
 - [Herramientas](#)
 - [Reportes](#)
 - [Área de Consola](#)
5. [Descripción del Lenguaje](#)

Objetivo General

Aplicar los conocimientos sobre la fase de análisis léxico y sintáctico de un compilador para la construcción de una solución de software.

Objetivos Específicos

- Que el estudiante aprenda a generar analizadores léxicos y sintácticos utilizando las herramientas de JFLEX y CUP.
- Que el estudiante aprenda los conceptos de token, lexema, patrones y expresiones regulares.

- Que el estudiante pueda realizar correctamente el manejo de errores léxicos.
- Que el estudiante sea capaz de realizar acciones gramaticales utilizando el lenguaje de programación JAVA.

Descripción General

El curso de Organización de Lenguajes y Compiladores 1 perteneciente a la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala requiere de usted como conocedor de la construcción de analizadores Léxico y Sintáctico crear un sistema que sea capaz de realizar operaciones aritméticas y estadísticas además de poder generar diversos gráficos a partir de una colección de datos.

Entorno de Trabajo

Editor

El editor será parte del entorno de trabajo cuya finalidad será proporcionar ciertas funcionalidades, características y herramientas que serán de utilidad para el usuario. La función principal del editor será el ingreso del código fuente que será analizado. Queda a discreción del estudiante el diseño.

Funcionalidades

- **Nuevo archivo:** El editor debe tener la capacidad de crear archivos en blanco, el cual podrá ser editado en una pestaña que tiene el nombre del archivo.
- **Abrir archivo:** El editor debe tener la capacidad de abrir archivos con las extensiones .df, cuyo contenido se deberá mostrar en el área de entrada en una nueva pestaña con el nombre del archivo.
- **Guardar:** El editor debe tener la capacidad de guardar el estado del archivo en el que se estará trabajando.
- **Eliminar pestaña:** Cada pestaña puede ser cerrada en cualquier momento. Si los cambios no se han guardado, se descartan.

Características

- **Múltiples pestañas:** Se podrán crear nuevas pestañas con la finalidad de ver y abrir los archivos de prueba en la aplicación. Para cada pestaña corresponde un archivo.

Herramientas

- **Ejecutar:** Se envía la entrada de la pestaña actualmente seleccionada al intérprete con la finalidad de realizar el análisis léxico, sintáctico y la ejecución de instrucciones.

Reportes

- **Reporte de Tokens:** Se mostrarán todos los tokens reconocidos por el analizador léxico.
- **Reporte de errores:** Se mostrarán todos los errores léxicos y sintácticos encontrados.
- **Reporte de Tabla de Símbolos:** Se mostrarán todas las variables y arreglos declarados.

Área de Consola

En la consola de salida se mostrarán los resultados, mensajes y todo lo que sea indicado en el lenguaje. Tiene como restricción el no ser editable por el usuario y únicamente puede mostrar información.

Descripción del Lenguaje

El lenguaje DataForge presenta características como ser case insensitive, manejo de encapsulamiento, comentarios, tipos de datos, declaraciones de variables, estructuras de datos como arreglos, operaciones aritméticas, funciones estadísticas, impresión en consola y funciones de graficación.

A continuación, se muestran espacios de texto con los comandos incluidos en el proyecto DataForge, junto con ejemplos de cada uno:

Declaración de Variables

- **Tipo Char:**

```
var:char[]::cadena <- "Hola Mundo" end;
```

- **Tipo Double:**

```
var:double::numero <- 3.1416 end;
```

Operaciones Aritméticas

- **Suma:**

```
var:double::suma <- SUM(5, 10) end;
```

- **Resta:**

```
var:double::resta <- RES(10, 5) end;
```

- **Multipliación:**

```
var:double::multiplicacion <- MUL(3, 4) end;
```

- **División:**

```
var:double::division <- DIV(10, 2) end;
```

- **Módulo:**

```
var:double::modulo <- MOD(10, 3) end;
```

Funciones Estadísticas

- **Media:**

```
var:double::media <- Media(@arreglo) end;
```

- **Mediana:**

```
var:double::mediana <- Mediana(@arreglo) end;
```

- **Moda:**

```
var:double::moda <- Moda(@arreglo) end;
```

- **Varianza:**

```
var:double::varianza <- Varianza(@arreglo) end;
```

- **Máximo:**

```
var:double::maximo <- Max(@arreglo) end;
```

- **Mínimo:**

```
var:double::minimo <- Min(@arreglo) end;
```

Impresión en Consola

- **Imprimir Expresiones:**

```
console::print = "El resultado es: " numero end;
```

- **Imprimir Arreglos:**

```
console::column = "Valores del Arreglo" -> @arreglo end;
```

Funciones de Graficación

- **Gráfica de Barras:**

```
graphBar(  
  titulo::char[] = "Gráfica de Barras" end;  
  ejeX::char[] = ["Enero", "Febrero", "Marzo"] end;  
  ejeY::double = [20, 30, 40] end;  
  tituloX::char[] = "Meses" end;  
  tituloY::char[] = "Ventas" end;  
  EXEC graphBar end;  
) end;
```

- **Gráfica de Pie:**

```
graphPie(  
  label::char[] = ["Producto A", "Producto B", "Producto C"] end;  
  values::double = [50, 30, 20] end;  
  titulo::char[] = "Distribución de Productos" end;  
  EXEC graphPie end;  
) end;
```

- **Gráfica de Línea:**

```
graphLine(  
  titulo::char[] = "Gráfica de Línea" end;  
  ejeX::char[] = ["Semana 1", "Semana 2", "Semana 3"] end;  
  ejeY::double = [15, 20, 25] end;  
  tituloX::char[] = "Semanas" end;  
  tituloY::char[] = "Producción" end;  
  EXEC graphLine end;  
) end;
```

- **Histograma:**

```
Histogram(  
  titulo::char[] = "Histograma de Edades" end;
```

```

values::double = [20, 25, 30, 25, 20, 30] end;
EXEC Histogram end;
) end;

```

Estos son algunos de los comandos y ejemplos que se incluyen en el proyecto DataForge. Puedes adaptarlos y modificarlos según las necesidades de tu proyecto.

Reportes

Los reportes son parte fundamental de DataForge, incluyendo tablas de tokens, errores y símbolos, y deben ser generados en formato HTML después de cada ejecución.

Reporte de Tokens

Lexema	Tipo	Línea	Columna
1 PROGRAM	Palabra Reservada	1	1
2 var	Palabra Reservada	2	1
3 :	Dos Puntos	2	4
4 double	Tipo de Dato	2	5
5 ::	Doble Dos Puntos	2	11
6 numero	Identificador	2	14
7 <-	Asignación	2	21
8 3.1416	Número Decimal	2	24
9 end	Palabra Reservada	2	30
10 ;	Punto y Coma	2	33
11 END	Palabra Reservada	3	1
12 PROGRAM	Palabra Reservada	3	5

```

### Reporte de Errores

```

```

```text

```

Tipo	Descripción	Línea	Columna
1 Léxico	Carácter no reconocido '\$'	4	12
2 Sintáctico	Se esperaba ';' pero se encontró 'var'	5	1

```

Tabla de Símbolos

```

```
```text
```

	Nombre	Tipo	Valor
	<hr/>		
1	numero	double	3.1416
2	arreglo	double[5]	[1,2,3,4,5]
3	cadena	char[]	"Hola Mundo"

Estas tablas pueden ser generadas en formato HTML para una visualización más clara y estética en un navegador web. Puedes ajustar los estilos y el diseño según tus preferencias.