

JAX

El analizador léxico es conocido como escáner, pues su principal funcionalidad es analizar los lexemas sobre algún patrón definido. Además, su principal funcionalidad es tomar la secuencia de caracteres o símbolos del alfabeto del lenguaje y ubicarlas dentro de las diferentes categorías comúnmente conocidas como unidades léxicas, que son empleadas por el analizador gramatical para determinar que lo escrito en el programa fuente este correcto o no gramaticalmente, de aquí que algunas de estas unidades léxicas sean rechazadas o removidas por ser un comentario o un espacio en blanco.

¿Qué es el analizador léxico jax?

Jax es un compilador léxico creado en lenguaje Java, que genera un escáner a partir de expresiones regulares que existen por defecto en un archivo de java.

¿Cómo se utiliza el jax?

Es encargado de procesar las expresiones regulares en el cual se genera un fichero Java que pueda ser compilado por Java y así crear el escáner.

Los escáneres generados por Jax tienen entradas de búfer de tamaño arbitrario, y es al menos más conveniente para crear las tablas de tokens, Jax utiliza solo 7 bits de caracteres ASCII, y no permite código Unario.

Jax

Código generado: Java.

No soporta entornos, está basado en expresiones regulares.

No soporta Unicode.

Ejemplo:

En esta fase solo vamos a identificar los tokens que hay en el código fuente de entrada y cada uno de sus tipos, no vamos a verificar la precedencia de operadores ni si la expresión aritmética es válida ya que ese es trabajo del analizador sintáctico.

Entonces asumamos un alfabeto con los siguientes lexemas y sus expresiones regulares:

🚦 Dígito: $[0-9]^+$

🚦 Operador Binario: $[*|/|+|-]$

Ahora definamos una clase que nos ayude a almacenar estos lexemas en una estructura de datos, llamaremos a esta clase “Token” y sería así:

```
public class Token {  
    public Tipos getTipo() {  
        return tipo;  
    }  
    public void setTipo(Tipos tipo) {  
        this.tipo = tipo;  
    }  
    public String getValor() {  
        return valor;  
    }  
    public void setValor(String valor) {  
        this.valor = valor;  
    }  
    private Tipos tipo;  
    private String valor;  
    enum Tipos {
```

```
NUMERO ("[0-9]+"),  
OPERADOR_BINARIO ("[*|/|+|-]");  
public final String patron;  
Tipos(String s) {  
    this.patron = s;  
}  
}  
  
}
```