



Procesadores del Lenguaje

Curso 2020-2021

Práctica 1.2: Análisis léxico: calculadora léxica independiente

1 Introducción

En esta segunda práctica, una vez familiarizados con la herramienta JFlex para construir analizadores léxicos, y habiendo desarrollado un primer proyecto desacoplado del analizador sintáctico, vamos a construir una calculadora básica únicamente con el analizador léxico

2 Especificación

En el ejemplo anterior vimos como implementar un reconocedor de expresiones matemáticas con números enteros, con posibilidad de agrupar con paréntesis, y lo extendimos con capacidad de reconocer elementos léxicos adicionales (comentarios, números reales, hexadecimales e identificadores de diferentes tipos).

En esta práctica se solicita modificar las acciones y patrones utilizados en el escáner (JFlex) para realizar las operaciones aritméticas básicas. Se recomienda utilizar el manual de la herramienta, y prestar atención en especial a la posibilidad de utilizar estados léxicos para refinar la especificación del analizador léxico

2.1 Expresiones simples

A. Expresiones simples: se solicita en primer lugar la capacidad de calcular los resultados con expresiones de únicamente dos operandos, separadas con “;”

- ⊙ $5+31$;
- ⊙ $-1*2$;
- ⊙ $-13*4$;

El resultado por pantalla debe mostrar el resultado de cada operación

Operacion: $5 + 31 = 36$

Operacion: $-1 * 2 = -2$

Operacion: $-13 * 4 = -52$

Sugerencia: se pueden utilizar estados léxicos para especificar un analizador con estados, de manera que en el arranque espere a leer el primer operando, después el operador, y después el segundo operando, y vuelva al estado inicial tras cada expresión. Pueden utilizarse variables auxiliares en el código del analizador para almacenar los valores de los operandos y operador de cada operación

A. Expresiones de múltiples operandos: se debe extender el analizador para tener capacidad de procesar expresiones con múltiples operandos y operadores como los ejemplos siguientes:

- ⊙ $5+31-2$;
- ⊙ $-1*2+2$;
- ⊙ $-13+10*4$;

De nuevo, el resultado por pantalla mostrará el resultado de cada operación (finalizada por “;”). Por simplicidad, la calculadora funcionará como una “máquina de pila” de dos operandos.

Es decir, asociando las operaciones de izquierda a derecha sin prioridad, de manera que el resultado de la última operación es el primer operando para la siguiente. En el ejemplo anterior la salida sería:

```
Operacion: 5 + 31 = 36
Operacion: 36 + -2 = 34
Resultado: 34
Operacion: -1 * 2 = -2
Operacion: -2 + 2 = 0
Resultado: 0
Operacion: -13 + 10 = -3
Operacion: -3 * 4 = -12
Resultado: -12
```

3 Entrega

Cada grupo debe entregar todo el contenido de su práctica en un único archivo comprimido – preferiblemente en formato ZIP–. El nombre del comprimido debe ser “pl_p1.2_grupo_XX”, donde XX son los primeros apellidos de cada componente del grupo.

Al abrir el archivo comprimido, debe verse un único directorio con el mismo nombre que el comprimido. Este directorio debe contener un proyecto eclipse completo y funcional, siguiendo la estructura del código de base. El código fuente del proyecto incluirá al menos los archivos con la especificación léxica del scanner que satisfagan los requisitos del enunciado.

NOTAS:

- *Se permite añadir nuevas clases Java si se considera necesario, y deberán incluirse en la entrega.*
- *Todos los proyectos son compilados por el corrector, de forma que NO se deben realizar modificaciones a los ficheros generados por JFlex, puesto que al compilar de nuevo estos cambios se perderán.*

Se adjuntará una breve memoria con consideraciones para el corrector. El formato deber ser TXT, DOC o PDF

Los dos criterios que primarán en la corrección son:

- Funcionalidad (principalmente). Todo debe funcionar sin errores y devolver la salida esperada.
- Buen uso de JFlex, que demuestre que se domina la herramienta.

En último lugar, cada grupo debe entregar ficheros de prueba, que contendrán entradas compuesta por código bien formado, que producirá una salida correcta del programa.