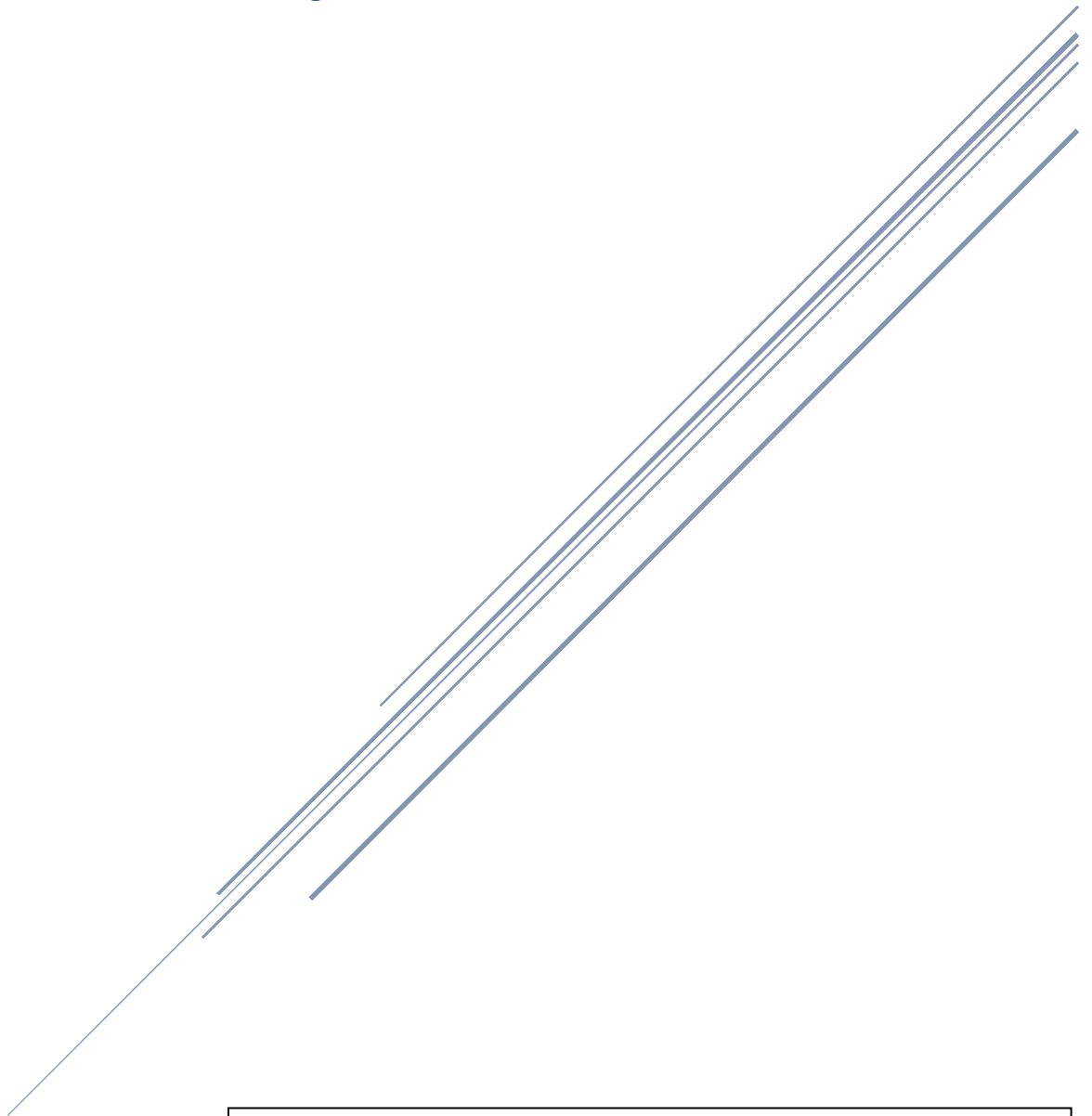


PROCESADORES DEL LENGUAJE

PRACTICA 1.2

Universidad Carlos III Madrid, Ingeniería Informática



IVAN AGUADO PERULERO – 100405871 JORGE SERRANO PÉREZ – 100405987



Contenido

1.- INTRODUCCIÓN	2
2.- PASOS SEGUIDOS	2
3.- CONSIDERACIONES PARA EL CORRECTOR	3
4.- CONCLUSIONES Y PROBLEMAS ABORDADOS	3

1.- INTRODUCCIÓN

En el presente documento se van a describir brevemente las decisiones de diseño tomadas frente a los problemas abordados en esta segunda parte de la práctica.

El objetivo de ella es construir una calculadora básica utilizando únicamente el analizador léxico desarrollado en la parte anterior mediante *JFLEX*.

2.- PASOS SEGUIDOS

Tras la implementación del analizador léxico de la primera parte nos disponemos a operar con los números reconocidos, ya sean enteros o decimales (incluso si están en notación científica).

Para ello hemos modificado el fichero *Driver.java*. En primer lugar, añadimos todos los símbolos reconocidos del fichero de entrada en el *arraylist* *símbolos*. A continuación, creamos otros dos *arraylist*, uno para almacenar los números con los que vamos a operar y otro para los operadores. A su vez inicializamos la variable *result* en la que posteriormente se irá almacenando el resultado de las operaciones.

Tras todo esto, mediante un bucle, recorreremos *símbolos* y vamos separando los operadores de los números. Aquí se ha decidido cambiar los “-” por “+” al guardarlos y almacenar los números en negativo si van precedidos de un “-”, de esta manera se simplifica nuestra tarea.

Una vez almacenados los símbolos que se necesitan para operar y cuando se detecta un “;” comienzan los cálculos. Para ellos hemos distinguido 3 casos diferentes, suma (en el que como hemos dicho se incluye la resta), multiplicación y división. En todos ellos se opera con el número de la posición actual y el siguiente. Tras obtener el resultado, este se almacena en la posición siguiente a la actual, para que la próxima vez actúe como el primer operando.

Una vez acabada una operación al completo los *arraylist* en los que almacenamos los números y operadores, al igual que la variable *result*, se resetean para poder incluir los símbolos necesarios para el siguiente cálculo.

Todo esto se repite hasta que se acaben los números y operandos almacenados, y en cada iteración se imprime por pantalla los resultados de cada par de operaciones. También se imprime al finalizar el resultado global.

Por último, para comprobar que nuestro código es válido y ejecuta correctamente todas las operaciones, se ha probado con el fichero de prueba *input.txt*, el cual tiene distintos

ejemplos de posibles cálculos con distintos operadores y números. Este se encuentra en la carpeta entregada.

3.- CONSIDERACIONES PARA EL CORRECTOR

Para que el código se ejecute correctamente se debe tener en cuenta lo siguiente:

- La calculadora solo funciona con números enteros y decimales, estos últimos pueden estar en notación científica.
- La operación solo se reconocerá si finaliza con el símbolo “;”.
- Las operaciones se realizan sin prioridades, es decir, de izquierda a derecha.
- Todos los símbolos de la practica anterior son reconocidos, pero si no son números u operadores se ignoran a la hora de hacer el cálculo.

4.- CONCLUSIONES Y PROBLEMAS ABORDADOS

Esta segunda parte de la práctica nos ha ayudado a entender mejor las utilidades de un analizador léxico, puesto que hemos tenido que utilizar el desarrollado en la primera parte para realizar las operaciones. Nos ha resultado un poco más fácil que la entrega previa, pues ya estábamos un poco más familiarizados con el entorno.

Aun así, nos ha frenado un poco las operaciones con números negativos. Este contratiempo ha sido debido a que nuestro analizador por decisión propia reconoce los números sin signo y estos los clasifica como operadores. Para que esto pueda continuar siendo así, tras probar varias implementaciones decidimos optar por la mencionada anteriormente, sustituir los signos “+” por “-” al almacenarlos como operadores y guardar los números precedidos por un “-” como negativos.

En conclusión, esta parte complementa a la parte anterior y nos ayuda a fijar los conocimientos que previamente aplicamos. En cuanto a la práctica completa, finalmente hemos conseguido desarrollar un proyecto desacoplado del analizador léxico que es capaz de realizar operaciones y de reconocer distintos *tokens* específicos.