

Procesamiento de Lenguajes (PL) Curso 2014/2015

Práctica opcional: traductor de expresiones hecho a mano

Fecha y método de entrega

La práctica debe realizarse de forma individual, como las demás prácticas de la asignatura, y debe entregarse a través del servidor de prácticas del DLSI **antes de las 23:59 del viernes 6 de febrero de 2015**.

Al servidor de prácticas del DLSI se puede acceder de dos maneras:

- Desde la web del DLSI (<http://www.dlsi.ua.es>), en el apartado “Entrega de prácticas”
- Desde la URL <http://pracdlsi.dlsi.ua.es>

Una vez en el servidor, se debe elegir la asignatura PL y seguir las instrucciones.

Evaluación y objetivo de la práctica

Esta práctica es opcional, aquellos alumnos que la realicen correctamente tendrán 1 punto adicional en la nota de prácticas en la asignatura (sí, es posible sacar un 11 en prácticas).

El objetivo de esta práctica es resolver un problema de traducción sin conocer y, por tanto, sin utilizar las técnicas que se estudiarán en la asignatura, de manera que se entienda la dificultad del problema y la necesidad de utilizar dichas técnicas en tareas similares.

Descripción de la práctica

La práctica consiste en diseñar un programa en Java que traduzca expresiones aritméticas sencillas a un lenguaje funcional. Por ejemplo, el programa debe traducir la expresión “2+3*4=” por “`suma(2,prod(3,4))`”.

El programa debe estar en una única clase llamada “`plp0`” y en un único fichero fuente que debe llamarse “`plp0.java`”. El programa debe leer un fichero de texto cuyo nombre será el único argumento que se le pase al programa desde la línea de comandos, y debe leer el fichero por líneas. En cada línea, el fichero tendrá una expresión, y el programa debe escribir por pantalla la traducción de cada línea.

Aunque el problema es bastante complicado, y en PL estudiaremos las técnicas necesarias para diseñar correctamente este traductor, en esta práctica tendrás que abordar el problema sin conocimientos de lo que se va a tratar en la asignatura, por lo que haremos las siguientes simplificaciones:

1. En la expresión solamente aparecerán números enteros sin signo de una sola cifra y los operadores + - * / =, siendo este último la marca del final de la expresión. Por tanto, no habrá espacios en blanco dentro de la expresión (no es necesario comprobarlo), ni tampoco paréntesis.

2. La expresión será siempre correcta, no contendrá errores de ninguna clase.
3. Los operadores $+$ y $-$ tienen la misma precedencia, y también los operadores $*$ y $/$ tienen la misma precedencia. Como en la mayoría de los lenguajes de programación, todos los operadores son asociativos por la izquierda (p.ej. " $2*3/4$ " es " $(2*3)/4$ "), y los operadores $+$ y $-$ tienen menos precedencia que los operadores $*$ y $/$, es decir, los operadores $*$ y $/$ se evalúan antes (p. ej. " $2+3*4$ " es " $2+(3*4)$ ").
4. El algoritmo para traducir (o evaluar) una expresión aritmética como las descritas es relativamente sencillo:
 - a) Se inicializan dos pilas, A y B
 - b) Se toma un carácter de la entrada
 - c) Si el carácter es un número, se apila en la pila A
 - d) Si el carácter de la entrada es un operador X , entonces
 - 1) si la pila B está vacía, se apila el operador X en la pila B
 - 2) si la pila B no está vacía, sea Y el operador en la cima de B
 - si X es de mayor precedencia que Y , se apila X
 - si X es de menor o igual precedencia que Y , se desapilan dos operandos de la pila A, siendo a_2 el operando que estaba en la cima de la pila A, y a_1 el operando que estaba debajo de a_2 antes de desapilar. Se desapila Y de la pila B, se apila en la pila A la traducción de la operación $a_1 Y a_2$, y se vuelve al paso d.1) sin cambiar el operador X
 - e) Si en la entrada aparece el operador $=$, mientras la pila B no esté vacía se van desapilando dos operandos de la pila A y un operador de B, y apilando en A la traducción. Finalmente, cuando B está vacía, se desapila de la pila A la traducción de toda la expresión y se termina el algoritmo
 - f) Volver al paso b)

Ejemplo

Por ejemplo, si el fichero tiene esto:

```
1+2=
3*4=
1+2*3=
1+2-3=
1*2+3=
1-2+4*5/6+3=
```

La salida por pantalla debe ser:

```
suma(1,2)
prod(3,4)
suma(1,prod(2,3))
resta(suma(1,2),3)
suma(prod(1,2),3)
suma(suma(resta(1,2),div(prod(4,5),6)),3)
```